

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# Bezdrátový systém senzorů a aktuátorů pro chytrou domácnost

## Wireless Sensor and Actuator System for a Smart Home

# Zadání diplomové práce

Student: **Bc. Vojtěch Petřík**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T041 Řídicí a informační systémy  
Téma: **Bezdrátový systém senzorů a aktuátorů pro chytrou domácnost**  
**Wireless Sensor and Actuator System for a Smart Home**  
Jazyk vypracování: čeština

## Zásady pro vypracování:

Diplomová práce se zabývá konstrukcí, návrhem a implementací subsystémů chytrého domu založených na specifickém bezdrátovém protokolu. Cílem diplomové práce je realizace modulů pro distribuované či decentralizované řízení vstupních i výstupních nodů.

V souhrnu je práce charakterizována těmito body:

1. Analýza stavu technologií chytrých domů.
2. Návrh a realizace senzorických nodů.
3. Návrh a realizace aktuátorů.
4. Návrh a realizace informačního nodu.
5. Analýza, návrh a realizace příslušného software.
6. Test hotového řešení.
7. Zhodnocení práce a závěr.

## Seznam doporučené odborné literatury:

- [1] HOROWITZ, Paul a Winfield HILL. *The Art of Electronics*. Cambridge University Press, 2015. ISBN-13: 978-0521809269.
- [2] KURNIAWAN, Agus. *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Packt Publishing, 2019. ISBN-13: 978-1789956870.
- [3] BEARD, Cory a William STALLINGS. *Wireless Communication Networks and Systems*. Pearson Education Limited, 2015. ISBN-13: 978-0133594171.
- [4] PRINZ, Peter a Tony CRAWFORD. *C in a Nutshell: The Definitive Reference*. O'Reilly Media, 2015. ISBN-13: 978-1491904756.
- [5] LEA, Perry. *Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security*. Birmingham, UK: Packt Publishing, 2018. ISBN-13: 978-1788470599.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



---

doc. Ing. Jiří Kozírek, Ph.D.  
*vedoucí katedry*

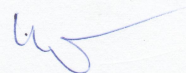


---

prof. Ing. Pavel Brandštetter, CSc.  
*děkan fakulty*

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020

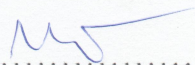


.....



Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 15. května 2020

.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Velké díky patří vedoucímu práce za trpělivost a ohromnou ochotu pomoci.

## **Abstrakt**

Tato práce se zabývá návrhem a realizací bezdrátových senzorů a aktuátorů pro chytrou domácnost. Jejím cílem je vytvořit v praxi použitelný systém s několika zařízeními umožňujícími vzájemný bezdrátový přenos dat. Pro základ je vytvořeno 6 různých zařízení pro zobrazení dat, měření kvality vzduchu v místnosti, bezdrátové tlačítko, bezdrátově ovládanou zásuvku, přijímač infračerveného signálu a dálkově ovládaný RGB LED pásek. Teoretický návrh se nejprve zabývá samotným pojmem chytré domácnosti a jejími vlastnostmi. V rámci této části je zahrnuto využití programovací prostředí a přidání platformy ESP32 pro programování. Další částí je interpretace softwarového řešení, kde je popsán použitý komunikační protokol ESP-NOW a návrh komunikace, spolu s vytvořenou knihovnou a popisem základních softwarových řešení jednotlivých zařízení. Následující část je věnována samotné tvorbě těchto zařízení - od fyzického vzhledu po vnitřní zapojení a tvorbu komponent. Zakončení je věnováno úspěchům a selháním při tvorbě této práce.

**Klíčová slova:** Chytrá domácnost, ESP32, automatizace

## **Abstract**

This work deals with the design and implementation of wireless sensors and actuators for the smart home. Its goal is to create a system that can be used in practice with several devices enabling mutual wireless data transmission. 6 different devices for data display, room air quality measurement, wireless button, wirelessly controlled socket, infrared signal receiver and remotely controlled RGB LED strip are created for the base. The theoretical proposal first deals with the very concept of a smart home and its properties. This section includes the programming environment used and the addition of the ESP32 programming platform. The next part is the interpretation of the software solution, which describes the used ESP-NOW communication protocol and communication design, together with the created library and a description of the basic software solutions of individual devices. The next part is devoted to the creation of these devices - from the physical appearance to the internal wiring and creation of components. Completion is devoted to successes and failures in creating this work.

**Keywords:** Smart home, ESP32, automatization

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>10</b>
<b>Seznam obrázků</b>	<b>11</b>
<b>Seznam tabulek</b>	<b>13</b>
<b>1 Úvod</b>	<b>14</b>
<b>2 IOT - Internet of things</b>	<b>15</b>
2.1 Chytré domácnosti . . . . .	15
2.1.1 Používané technologie . . . . .	15
2.2 Bezdrátový přenos dat . . . . .	16
<b>3 Vývojové prostředí</b>	<b>17</b>
3.1 Arduino IDE . . . . .	17
3.1.1 Přidání ESP32 do Arduino IDE . . . . .	17
<b>4 Interpretace softwarového řešení</b>	<b>21</b>
4.1 Použité názvosloví . . . . .	21
4.2 Topologie systému . . . . .	21
4.3 ESP-NOW . . . . .	22
4.4 Návrh komunikace . . . . .	23
4.5 Knihovna HAS.h . . . . .	25
4.6 Btn node . . . . .	26
4.7 Air node . . . . .	26
4.8 Out node . . . . .	27
4.9 Ir node . . . . .	29
4.10 RGB node . . . . .	30
<b>5 Návrh a realizace komponent</b>	<b>31</b>
5.1 Řídicí člen ESP32 . . . . .	31
5.1.1 Použitá platforma ESP32 Devkit v1 . . . . .	32
5.2 Napájení . . . . .	33
5.3 Krabičky pro zařízení . . . . .	34
5.4 Master node . . . . .	35
5.4.1 E-ink displej . . . . .	36
5.4.2 Návrh Master node . . . . .	37
5.4.3 Realizace Master node . . . . .	39
5.5 Air node . . . . .	39



5.5.1	SCD30 . . . . .	40
5.5.2	Návrh Air node . . . . .	41
5.5.3	Realizace Air node . . . . .	42
5.6	Btn node . . . . .	42
5.6.1	Návrh Btn Node . . . . .	43
5.6.2	Realizace Btn Node . . . . .	44
5.7	Out node . . . . .	44
5.7.1	Návrh Out Node . . . . .	44
5.7.2	Realizace Out Node . . . . .	45
5.8	Ir node . . . . .	45
5.8.1	IR demodulátor a ovladač . . . . .	46
5.8.2	Návrh Ir node . . . . .	47
5.8.3	Realizace Ir node . . . . .	48
5.9	RGB node . . . . .	48
5.9.1	Návrh RGB node . . . . .	48
5.9.2	Realizace RGB node . . . . .	50
<b>6</b>	<b>Test navrhnutého řešení</b>	<b>52</b>
<b>7</b>	<b>Závěr</b>	<b>54</b>
	<b>Literatura</b>	<b>56</b>
	<b>Přílohy</b>	<b>56</b>
<b>A</b>	<b>Seznam příloh</b>	<b>57</b>

## Seznam použitých zkratk a symbolů

AC	– Alternating Current
AP	– Access Point
ASCII	– American Standard Code for Information Interchange
BLE	– Bluetooth Low Energy
CCMP	– Counter Mode Cipher Block Chaining Message Authentication Code Protocol
DAC	– Digital Analog Converter
DC	– Direct Current
DPS	– Deska plošných spojů
GPIO	– General Purpose Input Output
GPS	– Global Positing System
IOT	– Internet Of Things
IR	– Infra Red
LEDT	– Light Emitting Diode
MAC	– Media Access Control
MIT	– Massachusetts Institute of Technology
NDIR	– Non Disperzive infrared
PCB	– Printed circuit board
PWM	– Pulse Width Modulation
RGB	– Red Green Blue
ROM	– Read Only Memory
SPI	– Serial Peripheral Interface
SRAM	– Static Random Access Memory
SSID	– Service Set Identifier
TFT	– Thin Film Transistor
UART	– Universal asynchronous receiver-transmitter
USB	– Universal serial bus

## Seznam obrázků

1	Vstup do Nastavení . . . . .	17
2	Přidání odkazu pro desky ESP32 . . . . .	18
3	Board Manager . . . . .	19
4	Instalace platformy ESP32 . . . . .	19
5	Vybrání aktuální desky . . . . .	20
6	Komunikace mezi zařízeními . . . . .	22
7	Vývojový diagram komunikace ESP-NOW . . . . .	24
8	Vývojový diagram Air node . . . . .	27
9	Vývojový diagram Out node . . . . .	28
10	ESP32 čip s integrovanou anténou . . . . .	31
11	Zapojení ESP32 Devkit v1 . . . . .	33
12	Modul měniče a nabíječky . . . . .	34
13	Návrh zadního krytu . . . . .	34
14	Návrh střední části . . . . .	35
15	Princip zobrazení na E-ink . . . . .	36
16	E-ink displej . . . . .	37
17	Návrh Master node . . . . .	38
18	Návrh DPS pro tlačítka . . . . .	38
19	Realizace Master node . . . . .	39
20	Senzorový modul SCD30 . . . . .	40
21	Princip NDIR . . . . .	41
22	Návrh Air node . . . . .	41
23	Realizace Air node . . . . .	42
24	Návrh Btn node . . . . .	43
25	Návrh DPS pro Btn node . . . . .	43
26	Realizace Btn node . . . . .	44
27	Blokové schéma Out node . . . . .	45
28	Realizace Out node . . . . .	45
29	Vnitřní zapojení demodulátoru . . . . .	46
30	Dálkový ovladač RGB LED pásku . . . . .	47
31	Návrh DPS Ir node . . . . .	47
32	Realizace Ir node . . . . .	48
33	Blokové schéma vnitřního zapojení RGB node . . . . .	48
34	Návrh ovládací části pro RGB LED pásek - schéma . . . . .	49
35	Návrh ovládací části pro RGB LED pásek - DPS . . . . .	49
36	Návrh ovládací části pro RGB LED pásek - rozmístění součástek . . . . .	50
37	Realizace ovládací části pro RGB LED pásek . . . . .	50

38	Realizace RGB node . . . . .	51
39	Zobrazení naměřených dat Air node . . . . .	52
40	Zobrazení dat na displeji . . . . .	53



## Seznam tabulek

1	Datová obálka protokolu ESP-NOW . . . . .	22
2	Datové pole . . . . .	23

# 1 Úvod

Tato diplomová práce je zaměřená na vytvoření senzorů a aktorů pro tvorbu chytré domácnosti. Součástí je vytvoření návrhu pro bezdrátově komunikující zařízení. S ohledem na variabilitu je vytvořeno 6 druhů: Master node - zobrazení dostupných zařízení v síti a dat z nich odesílaných, Air node - zařízení pro kontrolu kvality ovzduší (teplota, vlhkost, obsah CO<sub>2</sub>), Btn node a Out node - dálkově ovládaná zásuvka a Ir node s RGB node pro dálkově ovládané RGB LED pásy.

V dnešní době roste poptávka po levných a spolehlivých zařízeních v rámci Internetu věcí (IOT). Díky stále náročnějším požadavkům na snížení energetických výdajů domácností a automatizace či zlehčení prací, je vývoj těchto zařízení důležitý. Přístup k chytrým zařízením spojeným ať už pomocí kabelového vedení nebo bezdrátově v běžné domácnosti nemusí být jednoduchý, nicméně snaha pro co nejlepší optimalizaci tady vždy bude.

Jelikož je spousta možností ke zlepšení životních podmínek v domácnosti, zaměření v této práci je pro měření teploty, vlhkosti a obsahu CO<sub>2</sub> v místnosti a ulehčení ovládání RGB LED pásu a dálkové spínání zásuvky. Proto je tato práce postavena na tvorbě zařízení využívající jako hlavní řídicí člen s mikrokontrolér ESP32 a pro nízkou energetickou náročnost při přenosu dat (což prodlužuje výdrž u bateriově napájených zařízení) slouží komunikační protokol ESP-NOW.

V první části práce je k nalezení vysvětlení použití moderních technologií v rámci chytrých domácností a celkově vysvětlení pojmů jako je IOT a jejich problematika. Jednotlivé hlavní typy technologií využívaných při tvorbě chytrých domácností popřípadě využívané topologie jsou zde také.

Pro seznámení se s vývojovým prostředím využitým k snadné práci a programování našich zařízení slouží následující část. Zde je nejen objasněna možnost programování, ale i samotné přidání použité platformy do vývojového prostředí.

Ve třetí části se nachází samotné programové řešení. Jsou zde informace přímo k použitému komunikačnímu protokolu ESP-NOW a zároveň i popis vytvořené knihovny pro zjednodušení práce s daty využívanými v různých zařízeních naší práce. Najdeme zde i jednoduchý popis softwarového řešení některých zařízení.

Návrh a realizace komponent je čtvrtá část této práce. Hlavním rozbořem je popis použitých součástí a návrh s realizačním popisem jak vnější části zařízení, tak i vnitřního zapojení. Jednotlivé zařízení jsou zde popsány i z hlediska svého účelu.

Nakonec je v závěru zahrnut výstup této práce - celková funkčnost a další možné využití těchto zařízení v praxi.

## 2 IOT - Internet of things

Internet věcí (Internet of Things, IoT) je nový trend v oblasti kontroly a komunikace předmětů běžného využití mezi sebou nebo s člověkem a to zejména prostřednictvím technologií bezdrátového přenosu dat a internetu. [1]

Tyto zařízení, díky napojení na další systémy vyhodnocování dat umožní sběr velkého množství dat. Možností využití je spousta: přeprava, zdravotnictví, energetika, napojení na městské či domácí systémy. V dnešní době ekologie, kdy se lidé více a více zaměřují na ekonomickou stránku například vytápění či svícení světel, přibývá elektroinstalací v podobě tzv. "chytrých domácností".

Internet věcí není pouze možnost zapnout si na dálku topení na chatě či otevřít bránu pomocí telefonu, je to komplexní systém. Například, pokud máme s GPS modulem v telefonu propojenou domácnost, začneme-li se blížit po práci k domu, zapne se nám topení, otevře brána, kávovar začne připravovat kávu a spustí se naše oblíbená hudba.

Zároveň, díky sběru dat z celé domácnosti můžeme zjistit, že jeden z pokojů musí být vytápěn na vyšší teplotu než ostatní, neboť tam dochází k velkému úbytku tepla. Pokud v onom pokoji nemáme otevřené okno, může být řešením problému třeba zateplení. Sběr a vyhodnocení dat přináší nejen možnost ovládat věci na dálku, ale možnosti energetické optimalizace.

### 2.1 Chytré domácnosti

V dnešní době stále více používaný pojem Chytré domácnosti může trošičku zavánět inteligencí lidského mozku, nebo dnešních superpočítačů. Pravdou je, že domácnost má rozličné senzory a aktuátory, které komunikují nejen mezi sebou, ale jsou dostupné i přes internet (popřípadě jsou uloženy na cloudových službách).

Chytré domácnosti tedy zajišťují optimální prostředí pro komfort osob díky technologiím konstrukce, řídicích systémů a managementu. Jsou optimalizovány pro efektivní využití jak po energetické stránce, tak po stránce ekonomické. Díky těmto parametrům se snaží být i co nejvíce efektivní z hlediska působení na vnější prostředí (ekologický dopad).

Kromě ekonomického, energetického a ekologického dopadu se chytré domácnosti snaží co nejvíce zvýšit uživatelský komfort - zajištění bezpečí, zpříjemnění zábavy, snížení nákladů na provoz.

#### 2.1.1 Používané technologie

Pro označení Chytrá domácnost tedy uvažujeme domácnost, kde jsou různé senzory a aktuátory, které mezi sebou komunikují a navíc jsou připojeny přes internet. Na cloudovém úložišti mohou mít k zobrazení statistiky zaznamenaných dat ze senzorů, či možnost aktuálně měnit nastavení některých aktuátorů ručně.

Pro instalaci chytré domácnosti mohou sloužit dva druhy technologie: drátová a bezdrátová.

Drátová technologie se využívá hlavně u novostaveb, kdy je již při tvorbě elektroinstalace počítáno s vedením po drátech navíc. Při použití této technologie se stávající elektroinstalaci mohou nastat problémy při připojování - většinu je potřeba vytvořit nové drážky pro elektroinstalaci, což, pokud neprobíhá celková rekonstrukce elektroinstalace, může být značně namáhavé.

Bezdrátová technologie se používá primárně u již existujících staveb s vytvořenou elektroinstalací, kdy vhodnými senzory a aktuátory dochází k propojení pomocí bezdrátových technologií. Příkladem mohou být například chytré žárovky - kdy je možnost jednak ovládání pomocí klasického vypínače, ale i pomocí chytrého telefonu (díky připojení k bezdrátové síti).

Máme tedy k dispozici dvě rozdílné technologie přenosu dat. U daných systémů závisí ještě na topologii zapojení. Existují centralizované systémy, hybridní systémy a decentralizované systémy. [2]

- Centralizované systémy - u těchto systémů jsou vstupy (senzory) a výstupy (aktuátory) spojeny s centrálním řízením hvězdicově. Daná řízená oblast má tedy zapojeny všechny senzory a aktuátory na jedno místo. Příkladem tohoto zapojení mohou být například programovatelné automaty (PLC) v koupelně, kdy řídí větrání podle změřené vlhkosti. V chytrých domácnostech technologií centralizovaného řízení využívá například firma Loxone.
- Hybridní systémy využívají spojení centralizovaného řízení pro aktuátory a decentralizovaného řízení pro senzory. Senzory jsou tedy spojeny dohromady po sběrnici a předávají data do řídicích systémů, kde daný systém ovládá konkrétní aktuátory. Příkladem těchto systémů je systém Nikobus.
- Decentralizované systémy využívají datové sběrnice nejen pro senzory, ale i pro aktory. Každý senzor i aktor má tedy přítomen mikroprocesor, který dokáže data vhodně odesílat a přijímat. Příkladem instalací chytrých domácností je firma KNX.

Jednotlivé systémy mají své výhody i nevýhody. Centralizované (nebo Hybridní) systémy jsou plně závislé na řídicí jednotce. Pokud řídicí jednotka nefunguje, celý systém je odstaven. U decentralizovaných systémů je vždy spojena komunikace mezi určitým senzorem a aktuátorem. Při výpadku určitého senzoru či aktuátoru nefunguje pouze část soustavy, zbytek funguje dále beze změny.

## 2.2 Bezdrátový přenos dat

V rámci použitelnosti zařízení v chytrých domácnostech je možnost užít několik možností bezdrátového přenosu. V rámci použitelnosti to mohou být technologie jako Wifi, Bluetooth, ZigBee, IQRF a další.



### 3 Vývojové prostředí

Programování ESP32 je možno pomocí několika prostředí. Oficiální podporované vývojové frameworky přímo od výrobce jsou ESP-IDF a Arduino. Další programovací frameworky díky obrovské podpoře uživatelů například PlatformIO nebo Espruino.

#### 3.1 Arduino IDE

Toto velmi uživateli oblíbené prostředí díky své jednoduchosti mohou používat od začátečníků i pokročilí programátoři. Prostedí je využíváno pro programování různých platforem a mikrokontrolérů (Arduino, ESP, ARM, Atmel...). Nabízí možnost programování pomocí jazyka C nebo C++, přidání a správu knihoven, monitor pro komunikaci po sériové lince a spoustu dalšího.

Přestože programování ESP32 pomocí Arduino kódu je výrobcem přímo podporováno, pro využití programovacího prostředí Arduino IDE je potřeba tuto platformu nastavit.

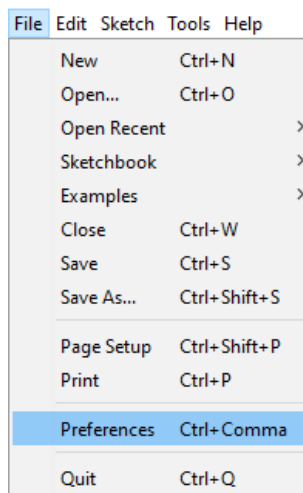
##### 3.1.1 Přidání ESP32 do Arduino IDE

Do nedávna bylo přidání platformy ESP32 do vývojového prostředí Arduino IDE zanečně složitější. Bylo potřeba stáhnout balíček s dat, vložit různé části na různá místa uloženého prostředí IDE, nainstalovat Python a spustit skripty pro kompilaci.

Dnes je instalace daleko jednodušší. Jde o přidání adresy pro aktualizace desky do správy desek a přidání desky ESP32 ve správci desek a o stažení aktuální verze se již nemusíme starat.[3]

1. Přidání odkazu do správce desek.

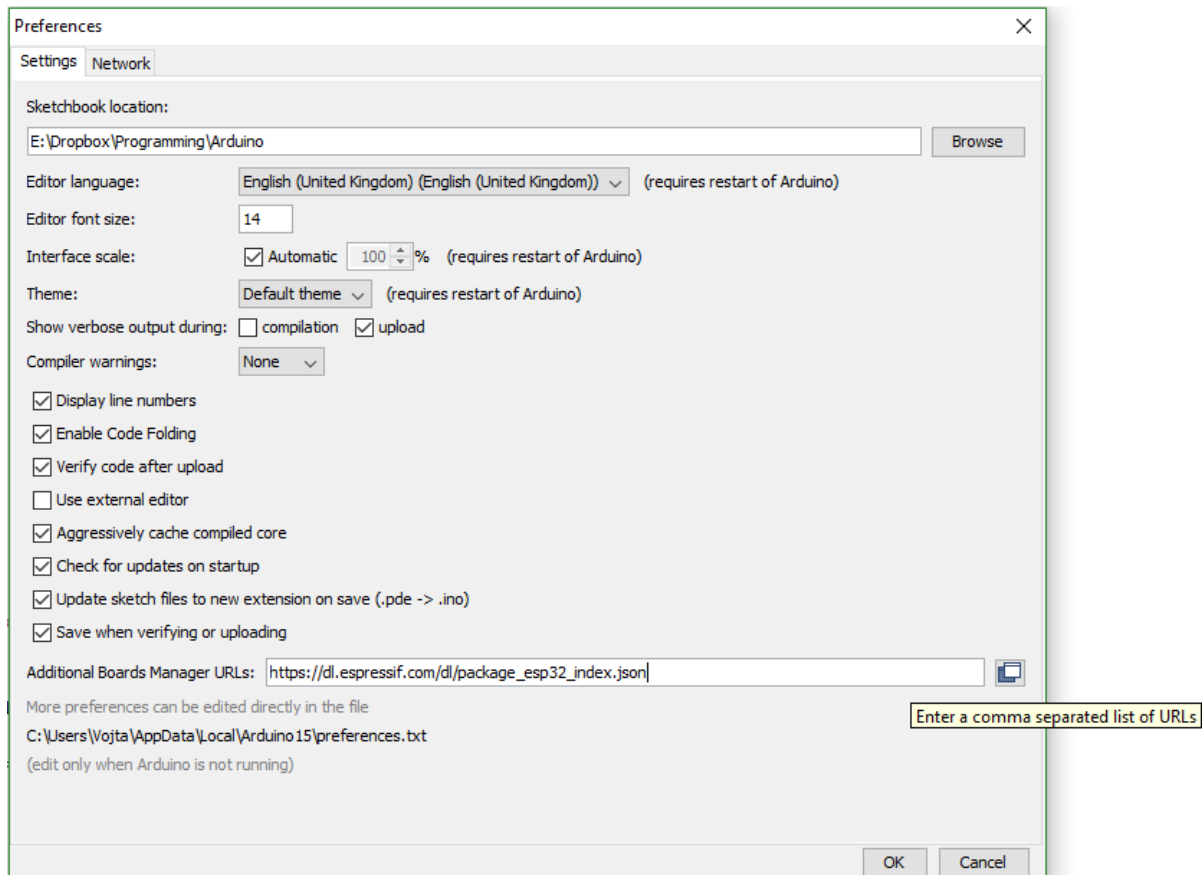
Po otevření Arduino IDE přejdeme na záložku File a otevřeme nastavení pomocí odkazu Preferences.



Obrázek 1: Vstup do Nastavení

Zde máme možnost nastavit několik voleb - nastavení velikosti a typu písma, zobrazení čísel řádku, ukládání při kompilaci a další. Nás zajímá řádek Additional Board Manager URLs - kde vkládáme odkazy na stahování dalších desek jiných výrobců pro použití v Arduino IDE.

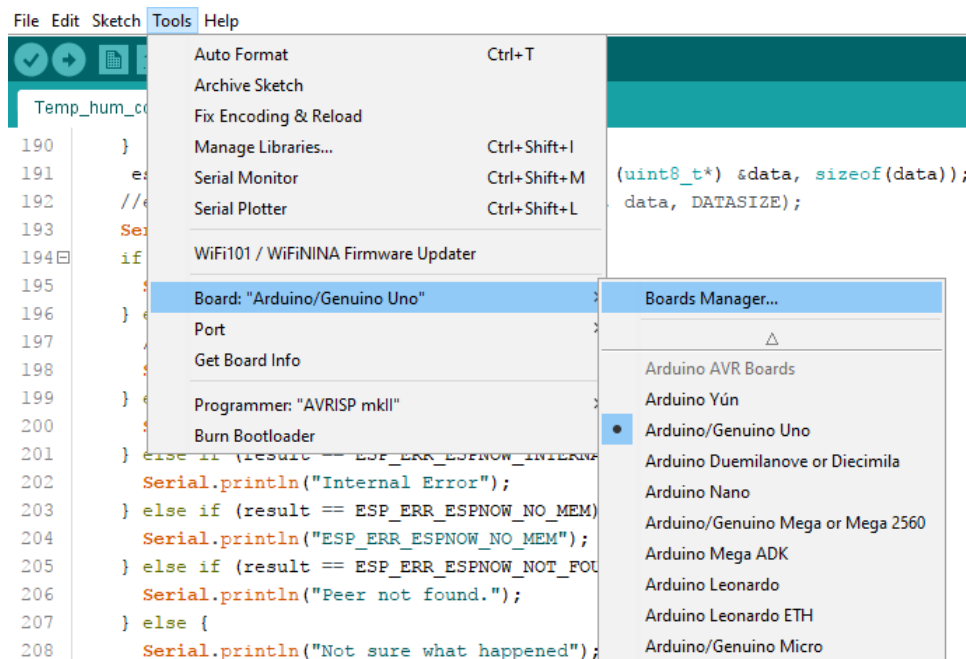
Pro použití platformy ESP32 je potřeba do řádku zadat odkaz [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) a zavřít okno s nastavením přes tlačítko OK.



Obrázek 2: Přidání odkazu pro desky ESP32

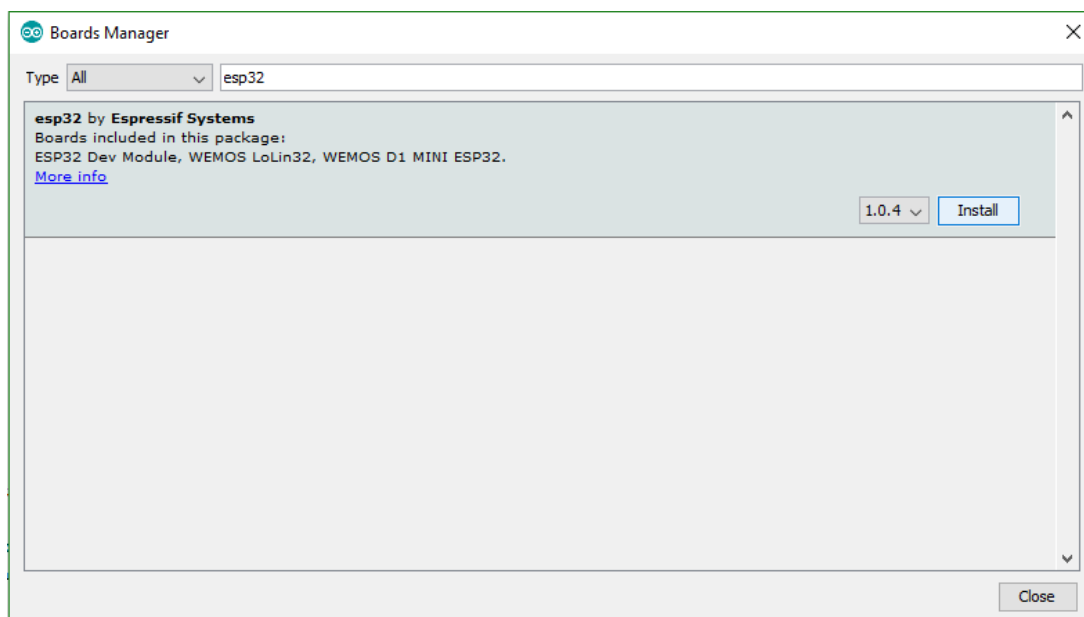
## 2. Přidání desky do Board Manageru

Po kliknutí na záložku Tools rozbalíme nabídku Board: kde nahoře otevřeme Boards Manager.



Obrázek 3: Board Manager

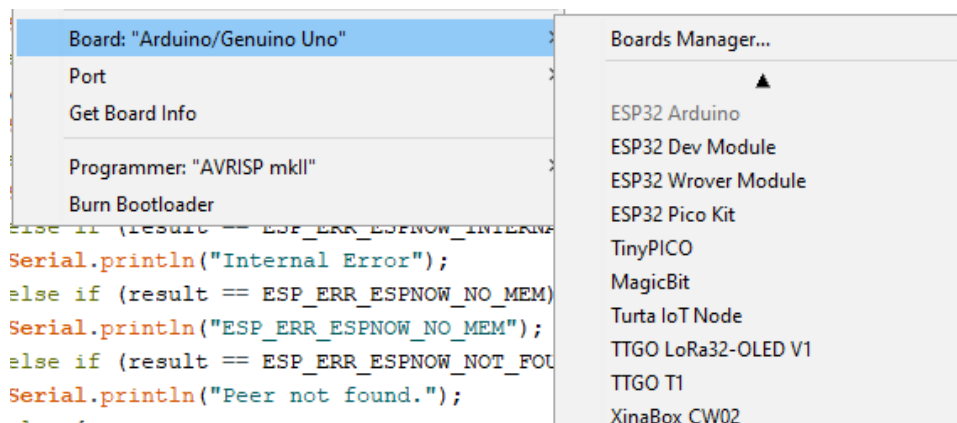
Zde zadáme název naší přidané platformy - ESP32 a nainstalujeme aktuální verzi.



Obrázek 4: Instalace platformy ESP32

### 3. Vybrání desky

Nyní již máme nainstalovány desky založené na mikrokontroléru ESP32. Pro programování naší desky vybereme nastavením desky ESP32 Dev Module.



Obrázek 5: Vybrání aktuální desky

Tímto způsobem již tedy máme připravené prostředí Arduino IDE pro programování na naší vybrané platformě ESP32. Již klasicky tedy můžeme použít všech výhod tohoto programovacího prostředí - ať už se jedná o možnost instalace oficiálních knihoven či monitor sériové linky.



## 4 Interpretace softwarového řešení

### 4.1 Použité názvosloví

V celé práci je použito a stále se opakuje několik názvů pro jednotlivé vyvíjené zařízení. Název node je přejat z angličtiny a je ve svém tvaru použit v kombinaci s vytvořenými názvy zařízení. Každý kus (každá krabíčka) je pojmenovaná dle svého primárního užití:

- Master node
- Air node
- Btn node
- Out node
- Ir node
- RGB node

Další aspekty použití po programové stránce jednotlivých zařízení jsou v této kapitole. Fyzická tvorba vzhledu, tvorba dalších komponent a jejich spojení je v kapitole 5. Návrh a realizace komponent.

### 4.2 Topologie systému

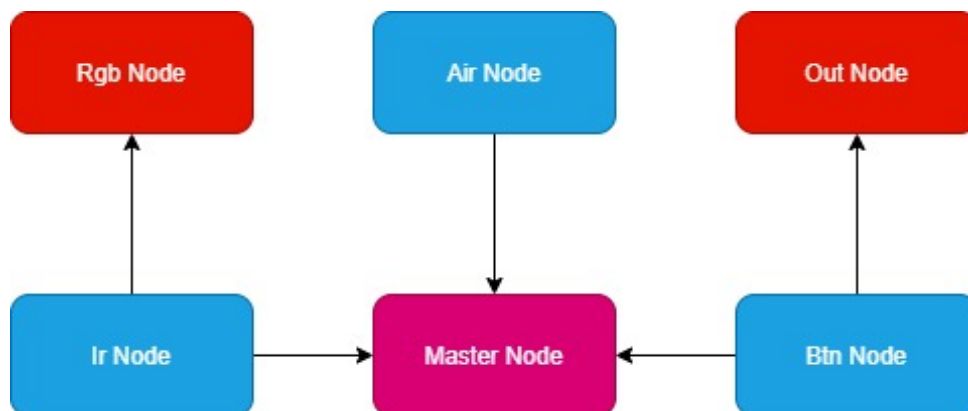
V této části práce bude představeno fungování po softwarové stránce. Jednotlivé zařízení (Master node, Air node, Btn node, Out node, RGB node a Ir node) vyváří mezi sebou komunikační vazby. Modré zařízení jsou vstupní, červené výstupní a fialová je řídící.

I když je použit název řídící (nebo Master node) u jednoho zařízení, stále se jedná o decentralizovaný systém. Komunikace tedy bude fungovat i nadále (pokud odstavíme Master node či kteroukoliv jinou).

Pro optimální přenos dat je použit integrovaný wifi modul a komunikační protokol ESP-NOW.

**Hlavní datový přenos tedy probíhá mezi:**

- Btn node -> Out node (bezdrátové řízení zásuvky)
- Ir node -> Rgb node (bezdrátové ovládání RGB Led pásku)
- Air node, Btn node, Ir node -> Master node (odesílání informací o změně stavu a jejich zobrazení)



Obrázek 6: Komunikace mezi zařízeními

Přestože je toto hlavní záměr komunikace, ve skutečnosti je navržená komunikace vytvořena stylem odesílání dat do všech dostupných zařízení. Tímto způsobem je tedy jednoduché připojit do sítě další komponenty, bez nutnosti přehrávat software ve všech zařízeních.

### 4.3 ESP-NOW

Tento protokol byl navržen firmou Espressif pro bezdrátovou low-power komunikaci bez nutnosti připojení k existující wifi síti. Esp-now protokol využívá 2,4 GHz wifi modul a technologicky obdobně jako bezdrátové myši odesílá malé pakety dat (255B) chráněné šifrovaným přenosem. Tento protokol se hojně využívá v dnešní době pro IOT, díky své nenáročnosti na elektrickou energii je vhodný pro rozličné senzory a aktuátory. [4]

Esp-now protokol je založen na standartu IEEE802.11 a pro kryptování využívá CCMP technologii. Díky tomu vytváří bezpečné komunikační řešení bez nutnosti připojení k existující wifi síti.

Esp-now protokol využívá specifické datové obálky (vendor-specific action frame) k posílání dat. Tento formát je následující:

Tabulka 1: Datová obálka protokolu ESP-NOW

MAC Header	Category Code	Organization Identifier	Random Values	Vendor Specific Content	FCS
24 bytes	1 byte	3 bytes	4 bytes	7 255 bytes	4 bytes

- MAC Header: MAC adresa dané nody
- Category Code: Toto pole je nastaveno na hodnotu 127 - specifikuje využití kategorie vendor.
- Organization Identifier: Zde je unikátní identifikátor (0x18fe34) která obsahuje první tři byte MAC adresy.

- Vendor Specific Content: Zde je pro datová obálka obsahující specifické pole odesílané zprávy:

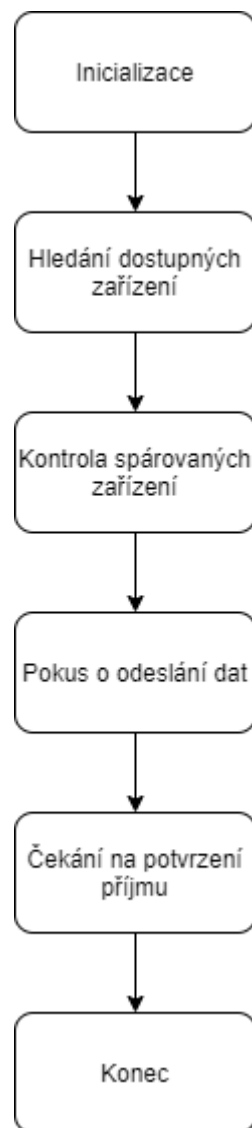
Tabulka 2: Datové pole protokolu ESP-NOW

Element ID	Length	Organization Identifier	Type	Version	Body
1 byte	1 byte	3 bytes	1 bytes	1 byte	0 250 bytes

- Element ID: Toto pole je nastaveno na specifickou hodnotu (221), indikující že jde o element vendor formátu.
- Length: Celková velikost následujících polí - Organization Identifier, Type, Version a Body.
- Organization Identifier: Zde je unikátní identifikátor (0x18fe34) která obsahuje první tři byte MAC adresy.
- Type: Zde je nastavena hodnota 4 - indikuje použití komunikačního protokolu esp-now.
- Version: Verze ESP-NOW protokolu.
- Body: Samotná odesílaná data.

#### 4.4 Návrh komunikace

Pro zahájení komunikace v síti mezi zařízeními je zapotřebí vykonat nejprve několik důležitých kroků:



Obrázek 7: Vývojový diagram komunikace ESP-NOW

- Po inicializaci (načtení knihoven, nastavení proměnných) dochází minimálně jednou k prohledání okolí k zjištění dostupných sítí.
- Každý modul vysílá specifickou adresu (SSID), podle které lze identifikovat zařízení komunikující v síti. Vždy je tvořena názvem - ESPNOW a MAC adresou daného zařízení. Díky tomu lze nalézt stejné moduly a zároveň porovnávat jejich MAC adresu s adresami v databázi.
- Dalším krokem při odeslání dat je kontrola párování - pokud již zařízení bylo dříve zjištěno, je uložena MAC adresa v paměti. Pokud tomu tak učiněno nebylo, dochází k zapsání.
- Po úspěšném párování přichází odeslání dat - na specifickou síť definovanou MAC adresou spárovaných zařízení dochází ke spojení a odeslání dat.

- Díky kontrolním mechanismům se vždy čeká na kontrolní odpověď zpět - zda dorazila data v pořádku a došlo k ukončení komunikace.

## 4.5 Knihovna HAS.h

Softwarová část je realizovaná ve vývojovém prostředí Arduino IDE. Pro použití dílčích částí kódu používaných zařízeními byla vytvořena knihovna HAS.h. Tato knihovna usnadňuje práci s převodem proměnných pro odesílání dat a hlavně pro centrální nastavení jednotlivých MAC adres použitých zařízení.

Tato knihovna obsahuje několik dílčích částí:

- **Inicializace knihovny**

Pro správnou funkci knihovny je třeba nastavit notificační LED diodu. Připojená LED dioda je již integrována na vývojové desce, nicméně díky této definici je umožněno používat notificační LED diodu i na jiném GPIO.

- **Práce se zařízeními**

Pro práci s jednotlivými zařízeními je zde definována struktura všech možných zařízení připojitelných do sítě. Tato struktura se skládá ze čtyř prvků - názvu zařízení, MAC adresy pro přijímání dat (Station - stanice), MAC adresy pro odesílání dat (AP - přístupového bodu) a stavu připojení definovaného jako bitová hodnota.

---

```
struct nodes {
    String name;
    uint8_t mac[6];
    uint8_t peer_mac[6];
    bool connected;
};
```

---

Důležitost této knihovny tkví v tom, že na jednom místě jsou definovány MAC adresy všech zařízení připojitelných do sítě.

Funkce pro práci se zařízeními getNode() slouží k vyhledání podle jména nebo MAC adresy.

- **Výpočetní funkce** V této části se nachází funkce pro výpočty a přepočty. Jelikož odesílaná data pomocí protokolu ESP-NOW jsou možné pouze ve formátu uint8\_t, jsou zde funkce pro přepočet větších čísel (typicky int a float) na části, které mohou být v pořádku odeslány. Pro rozdělení datového typu float slouží funkce integerPart() a decimalPart(). U datového typu int jsou použity funkce toUInt8() a toUInt8Mod().

Na straně příjemce je však potřeba tyto data poskládat nazpátek do srozumitelného formátu. K tomuto právě slouží funkce intFold() a floatFold().

- **Ostatní funkce** Poslední částí knihovny je definice pro využívání demodulovaných kódů infračerveného ovladače - pro jednoduchou správu například při změně ovladače.

## 4.6 Btn node

Jednoduchost v tomto ovládacím zařízení tkví v možnosti využití. Jednoduché tlačítko je připojeno na řídicí desku a nastaveno na přerušení. Díky tomu tedy data jsou odeslána pouze pokud je zmáčknuto tlačítko.

Celý princip funkce spočívá v tomto: v nekonečné smyčce je kontrolována proměnná `button_state`, která je změněna ve funkci vyvolané přerušením tlačítka. Kontrolní mechanismus proti překmitům tlačítka nebo náhodnému mačkání spočívá ve vytvoření dalšího přerušení.

Toto přerušení je časové. Při stisku tlačítka nastane vypnutí snímání přerušení tlačítka a zároveň se zapne časové přerušení. Po uplynutí času(5s) vyvolaná funkce znova nastaví přerušení pro stisk tlačítka a cyklus se může opakovat.

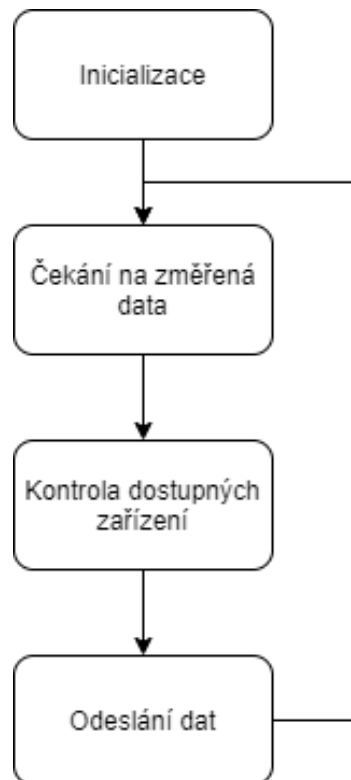
Funkce pro odeslání dat nejprve prozkoumá okolní síť, najde dostupná zařízení, pokud již nebylo provedeno tak provede párování a teprve poté dojde k odeslání dat. Tento cyklus trvá přibližně 2s, což je i reakční doba mezi zmáčknutím tlačítka a spuštěním spotřebiče.

## 4.7 Air node

Programově je toto zařízení jedno z jednodušších. Po inicializaci proměnných má za úkol jedinné: v periodicky se opakujícím tasku řízeném FreeRTOS systémem změřit data ze senzoru a poslat je připojeným zařízením.

Práci usnadňují knihovny pro sériovou komunikaci s použitým senzorem. Aby mohlo dojít k odeslání dat na všechny dostupné senzory, je potřeba vždy provést kontrolu dostupných zařízení před odesláním dat. Pokud by to nebylo provedeno, vždy by byly data poslány na zařízení připojené v inicializační části po startu programu.

Díky tomuto opatření je sice na jednu stranu umožněno spojit se se všemi aktuálními zařízeními v síti, na druhou stranu je to vykoupeno stejně jako u Btn node 2s zpožděním(toto zpoždění je způsobeno hledáním dostupných zařízení, popřípadě jejich párováním).



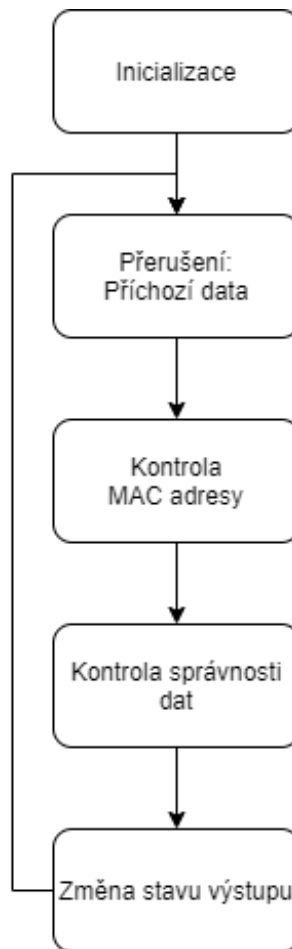
Obrázek 8: Vývojový diagram Air node

#### 4.8 Out node

Out node má za úkol jedinnou věc: zapínat nebo vypínat digitální výstup. Použití tohoto zařízení je pro bezdrátové ovládání zásuvky. Jako akční člen je použito relé spínané digitálním výstupem.

Pro správnou funkčnost je potřeba implementovat potřebné knihovny (WIFI.h pro obsluhu wifi modulu a práci s pakety, espnow.h pro využití komunikačního protokolu ESP-NOW a HAS.h pro využití funkcí výpočtů).

Způsob přijímání dat v následujícím diagramu je stejný u všech aktuátorů (RGB node). U senzorických zařízení (Air node, Btn node a Ir node) dochází při příjmu dat odeslaných z Master node po následné kontrole k odeslání aktuálního stavu daného zařízení místo změny stavu výstupu.



Obrázek 9: Vývojový diagram Out node

Nejdůležitější část kódu pro správný příjem dat je přijímací funkce `OnDataRecv`. Tato funkce je spuštěna přerušením příchozích dat a má za úkol zpracovat získaná data. MAC adresa odesílatele je rozdělena do pole, se kterým dále pracujeme.

Jsou zde dva kontrolní mechanismy pro odesílání falešných zpráv. První je kontrola MAC adresy zařízení (kontroluje funkce z vytvořené knihovny HAS - `control_nodes()`), kdy na začátku programu máme zadané MAC adresy zařízení, od kterých jsou data přijímána. Prakticky jsou přijímána jakákoliv data od kteréhokoliv zařízení - nicméně pouze přijatá data od námi definovaných (pomocí MAC adres) budou zpracována. Druhý kontrolní mechanismus je datová struktura - začátek přijaté zprávy musí obsahovat `*` (ASCII znak 42).

---

```

void OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len) {
    char macStr[18];
    snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
             mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr
             [5]);
  
```



```

Serial.print("\tLast Packet Recv from: "); Serial.println(has.getNode(
    mac_addr).name);
Serial.print("\tReceived data: ");
    for(int i =0; i< data_len;i++){
        Serial.print(data[i]); Serial.print(", ");
    }

if(has.control_nodes("Btn node", mac_addr)){
if(data[0] == 42){
    output_state = !output_state;
    //Serial.println(output_state);
    digitalWrite(OUTPUT_PIN, output_state);
    if(output_state)Serial.println("\n\tON");
    else Serial.println("\n\tOFF");
}
}
if(has.control_nodes("Master node", mac_addr)){
    if(data[0] == 42){
        sendData("Master node");
    }
}
}

```

---

Pakliže není MAC adresa shodná (Btn node nebo Master node) nebo data neobsahují na první pozici \*, dochází k ukončení kontroly a nastává čekání na další přerušení.

Přijatá data z Btn node mají za následek změnu výstupního stavu (zapnutí/vypnutí výstupu) pro ovládání relé. Master node výstup neovládá, pouze zjišťuje stav výstupu. Toho je dosíleno odesláním dat pomocí funkce `sendData("Master node")` se speciálním parametrem, který určuje validní odeslání dat aktuálního stavu výstupu na dané zařízení (konkrétně Master node).

## 4.9 Ir node

Toto senzorické zařízení snímá příchozí data z demodulátoru. Validace dat je kontrolována pomocí knihovny HAS.h.

Ve 100ms smyčce je prováděna kontrola, zda jsou dekodována demodulátorem nějaká příchozí data. Pokud jsou data přijata, dochází k porovnání s klíčem nastaveným v knihovně HAS.h a přiřazením ID daného signálu k odeslání. To je nastaveno po vyhledání zařízení v síti jako odesílaná zpráva. V RGB node dojde po přijetí dat k dekodování zprávy a přijatý signál z ovladače bude dále zpracován.

## 4.10 RGB node

Ovládání RGB (popřípadě RGBW) LED pásku je záležitostí tohoto zařízení. Pomocí PWM je nastaven průchod proudu řídícími tranzistory a tím pádem i regulace jasu RGB LED pásku. V rámci možnosti připojení RGB/RGBW je nutno v programu definovat daný výběr pomocí makra. Díky tomu je variabilita použití u těchto dvou typů pásků vyšší.

Nekonečná smyčka programu jednoduše funguje jako stavový automat pro jednotlivé nastavení RGB LED pásku. Data o změně stavu (zapnutí nebo vypnutí celého pásku, nastavení módů flash/fade/smooth, intenzita jasu či rychlost blikání) jsou nastaveny vždy při příjmu dat z Ir node po zmáčknutí tlačítka ovladače.

Jednotlivé možnosti svícení a blikání nastavené pro RGB LED pásek jsou tyto:

- Lights off - zhasnutí RGB LED pásku
- White Light - rozsvícení bílého světla (u RGBW pásku dochází k rozsvícení pouze bílé LED diody, u RGB je bílá barva vytvořena rozsvícením červené, zelené a modré barvy zároveň)
- RGB Changing - změna barev červená/zelená/modrá za sebou, s možnou změnou intervalu rychlosti změny
- RGB Fading - postupná změna intenzity jednotlivých barev se změnou barev. Každá barva samostatně nabývá na intenzitě a zase klesá, přičemž po klesnutí na 0 dojde ke změně barvy a změna intenzity se opakuje. V tomto případě je možnost změny rychlosti přechodu mezi úrovněmi jednotlivých barev.
- RGB Smoothing - Plynulý přechod barev skrze celé spektrum barev. K tomuto dochází pomocí přepočtu modelu barev HSI [5]. Pro změnu rychlosti prolínání barev také možno využít ovladač.
- Simply Color - nastavení barvy světla bez efektů blikání nebo prolínání barev. Regulace intenzity světla.

Pro běžného uživatele tedy skýtá RGB node spousty možností pro nastavení barev. Velikou výhodou tohoto vyvíjeného systému je fakt, že pomocí jednoho zařízení (Ir node) je možno zároveň ovládat několik RGB node rozmístěných v dosahu signálu.

## 5 Návrh a realizace komponent

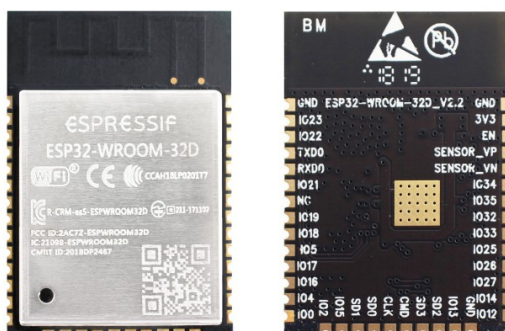
V této kapitole jsou k nalezení hlavní použité komponenty pro tvorbu jednotlivých senzorů a aktuátorů při výsledné práci.

Při realizaci hardware bylo zvoleno použití vlastních krabiček vytisknutých na 3D tiskárně. Díky této variabilitě bylo možno vytvořit osobitý design. Návrh krabiček se sestává ze 3 hlavních prvků - zadního krytu, středové části a horního krytu. Celkový návrh byl proveden v programu Inventor 2018.

Zároveň byla potřeba vytvořit pro některé nody DPS s ovládacími prvky (uchycení pro tlačítka, Ir demodulátor...).

### 5.1 Řídicí člen ESP32

Hlavní součástí pro vytvořené senzory a aktuátory byl vybrán použitý čip ESP32. Tento velice úspěšný nástupce čipu ESP8266 firmy Espressif Systems má několik nesporných výhod. Oproti svému předchůdci disponuje nejen integrovaným WIFI modulem, ale i modulem Bluetooth a spoustou dalších vylepšení.



Obrázek 10: ESP32 čip s integrovanou anténou

Esp32 je dvoujádrový čip vytvořená pomocí 40nm technologie. Je navrhnut pro dosažení nejlepších možných podmínek v rámci low-power systémů. Jako takový je vyvinut pro použití v nositelné elektronice a IOT aplikací. Samotný čip nepotřebu mnoho přídavných periférií nebo PCB. Obsahuje okolo 20 externích komponent.[6]

Specifikace:

- Dvojjádrový procesor 240MHz Tensilica LX6.
- Integrovaná 440kB ROM
- Integrovaná 520kB SRAM
- Interní 8MHz oscilátor s kalibrací

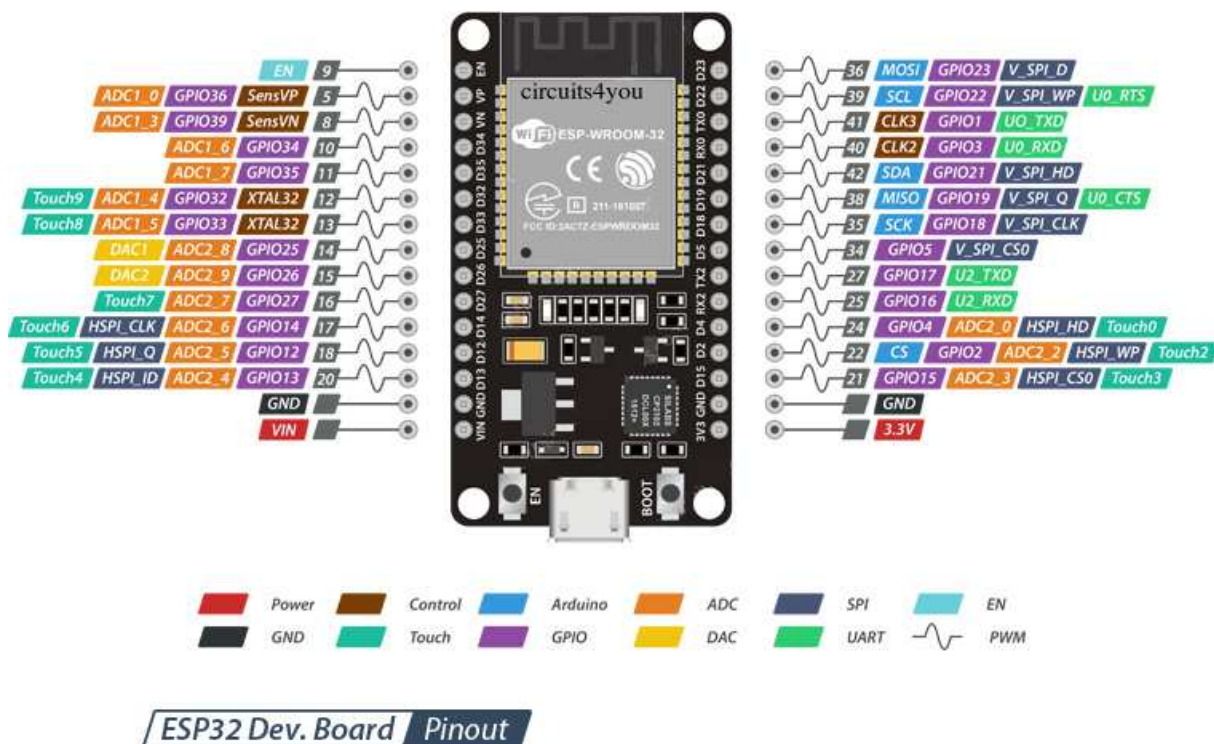
- 34 programovatelných GPIO pinů (s rozhraními jako SPI, I2C, DAC převodníky, UART a dalšími)
- Integrovaný Wifi vysílač a přijímač podporující standardy IEEE 802.11 b/g/n/e/i v 2,4 GHz pásmu.
- Možnost zabezpečení Wifi provozu pomocí WEP, WPA/WPA2 PSK/Enterprise
- Integrovaný dvoumódový Bluetooth (Classic i BLE)
- Zabudovaná PCB anténa
- Pracovní napětí 2,3 V až 3,6 V
- Pokročilý Low-power Management (správa nízké spotřeby) umožňující přepínání mezi pěti různými módy spotřeby a tím výrazně zvýšit dobu provozu zařízení napájené pouze z vlastní baterie.
- Provozní teplota od  $-40^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$ .

#### 5.1.1 Použitá platforma ESP32 Devkit v1

Pro jednoduché použití byl vybrán Devkit v1. Obsahuje ESP32 čip, 3,3V převodník napájecího napětí a CP2102 - převodník USB <-> UART s maximální přenosovou rychlostí 921600 baud. Vše je na jedné PCB desce připraveno k použití.

Výhoda použití této platformy spočívá v celém konceptu tvorby vyvíjených zařízení, a to modularitě jednotlivých částí pro vnitřní zapojení. Díky připraveným pinům je možno již jednoduše připojit senzory a výstupní zařízení.

Programování pak probíhá pomocí připojení micro USB kabelu a vytvoření sériové komunikace mezi převodníkem a samotným mikrokontrolérem ESP32.



Obrázek 11: Zapojení ESP32 Devkit v1

Jednotlivé piny jsou označeny podle svých možností zapojení - přes klasické programování pomocí ESP-IDF (GPIO piny jsou označeny šedým čtverečkem s číslem pinu) po programování pomocí Arduino IDE (Fialové políčka s číslem GPIO podle zavedeného bootloderu).

## 5.2 Napájení

Napájení jednotlivých modulů je rozdílné podle typu použití. Zařízení využívající napájení pro správnou funkci externích zařízení většího napětí jsou dvě - Out node a RGB node. Ostatní využívají bateriové napájení pro možnost přenosu a umístění zařízení.

Out node spíná zásuvku, a proto je přivedeno zde síťové napětí 230V. Toto napětí je pomocí malého pulzního zdroje ve formě modulu usměrněno a zmenšeno na provozní napětí 5V (je pouze pro napájení řídicího členu a spínacího relé), proto maximální proud je 500mA.

RGB node řídí napájení RGB LED pásku, a proto je externím zdrojem přivedeno napětí 12V. Toto napětí je dále pomocí měniče zmenšeno na 5V, které již využívá samotný řídicí obvod pro svou činnost.

Napájení ostatních node je vytvořeno pomocí li-ion baterie a modulu s měničem a nabíječkou baterie. Díky tomuto řešení je možnost dané node přenášet a umístit různě po místnosti. Měnič pro napětí je použitý MT3608. Díky svým vlastnostem (až 97% účinnost)[7] se skvěle hodí pro využití v low-energy systémech.

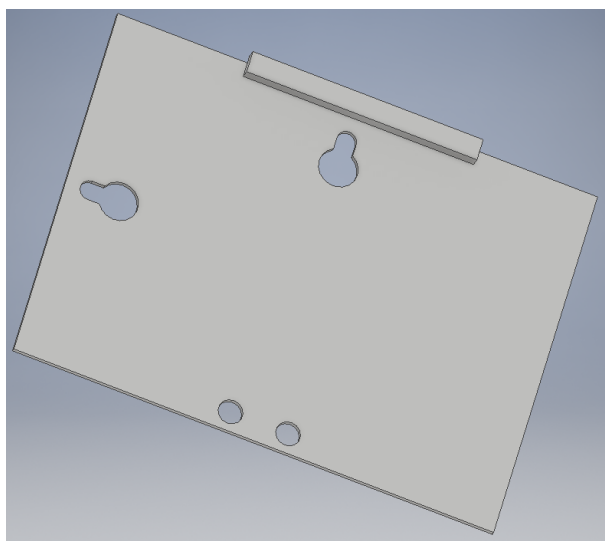


Obrázek 12: Modul měniče a nabíječky

### 5.3 Krabičky pro zařízení

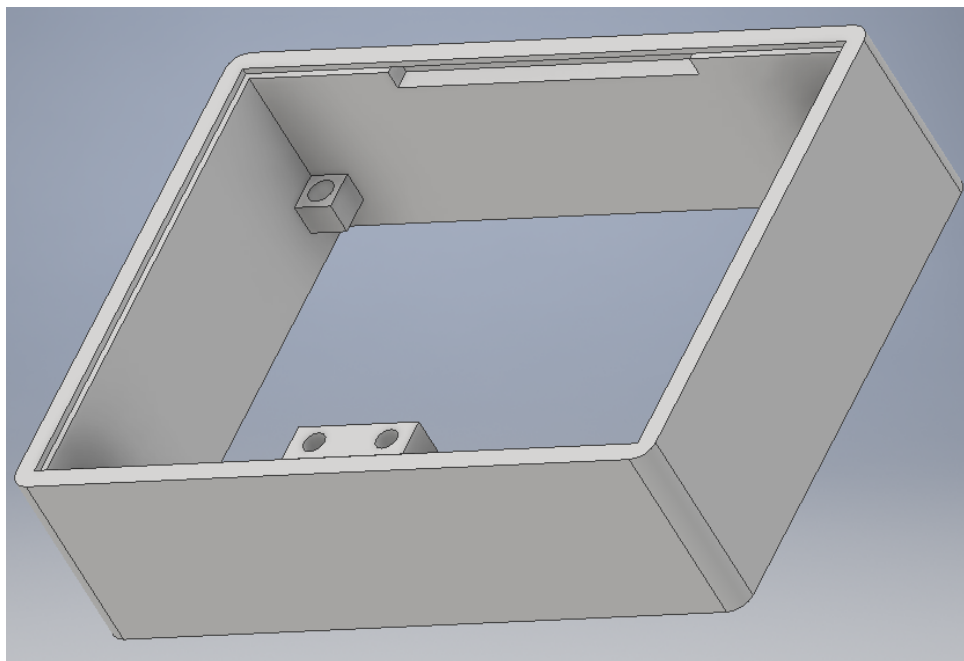
Návrh a design jednotlivých komponent pro vytvoření zařízení ve stejném (obdobném stylu) je vytvořen v modelovacím nástroji Autodesk Inventor 2018. Každá krabička se skládá ze tří hlavních částí - horního víčka, středové části a zadního krytu.

V rámci univerzálnosti jsou vytvořeny tyto části pro spojování šrouby. Zadní kryt je u všech krabiček stejný, zde nebylo potřeba vytvářet žádné variace. Přestože jsou zde na dvou stranách vytvořeny dírký pro zavěšení, je stejný zadní kryt použit i u zařízení, kde se nepředpokládá jejich umístění zavěšením (RGB node či Out node). Na druhou stranu, je pravděpodobné využití této nesporné výhody u Air node nebo Master node, kde přeci jenom dochází k možnosti měřit či zobrazovat data v určité výšce.



Obrázek 13: Návrh zadního krytu

Střední část tvoří obdélníkový tvar s vnějšími zakulacenými rohy. Ve vnitřní části jsou pro uchycení horního víčka vytvořeny struktury. Zatímco vrchní víčko je shora připevněno na krabičku, zadní kryt má vnitřní výřez pro zapadnutí. Tato část je již zvláště upravena pro vyvedení konektorů v RGB node a Out node, kde jsou z boku krabičky vyvedeny otvory a připraveny konektory (RGB node), vypínač a průchodky s kabely (Out node). Celý střední díl má na výšku 30mm. Vnější rozměry jsou 106\*81mm, takže velikostí je stvořena k dokonalému uložení vnitřních komponent, zároveň s minimalistickou velikostí pro uložení v místnosti.



Obrázek 14: Návrh střední části

Vrchní víčko je více variabilní. Jsou vytvořeny 3 rozdílné návrhy - pro Air node, Master node a Btn node (stejný návrh je použit u ostatních zařízení). Air node je charakteristický otvory pro výměnu vzduchu (možnost měřit teplotu, vlhkost a obsah  $\text{CO}_2$ ). Master node má výřez pro displej a ovládací tlačítka a Btn node spolu s ostatními zařízeními má jednoduché tlačítko umístěné ve středu vrchního víčka.

## 5.4 Master node

Vcelku nejkomplexnější zařízení ze všech navrhnutých zařízení. Obsahuje e-ink displej pro zobrazení informací a tlačítka pro pohyb v kontextovém menu. Účelem je vytvořit přístupový bod ke všem dostupným zařízením v síti a zobrazovat jejich nastavení a přenášené informace.

Tato node patří mezi zařízení využívající pro své napájení integrovanou li-ion baterii. Pro ovládání a pohyb v menu slouží tlačítka připevněna na desce plošných spojů.

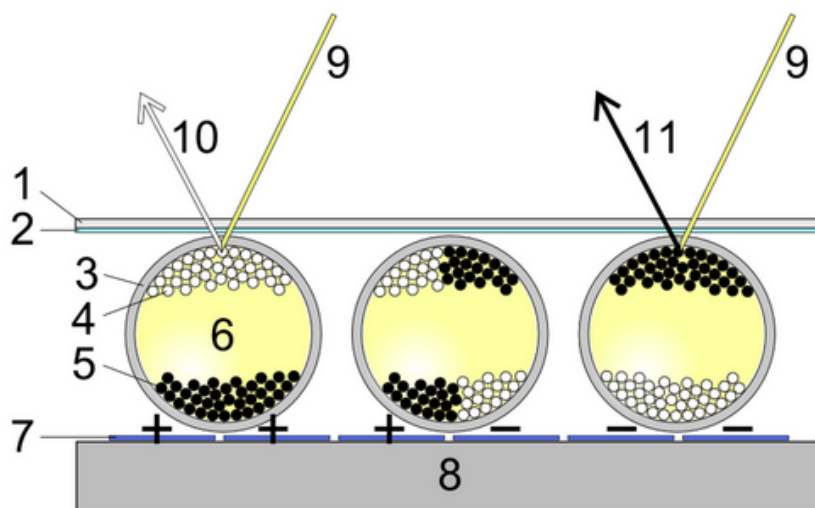
#### 5.4.1 E-ink displej

Tento typ displeje je již dlouhou dobu známý, nicméně své oblibě dosáhl o mnoho let později. První myšlenky pasivního displeje vznikly na MIT Media Lab v roce 1996, přičemž od roku 1997 vznikla firma E ink Corporation zabývající se tímto typem displeje. Masové nasazení této technologie probíhá ve čtečkách elektronických knih.

Obrovskou výhodou této technologie je její energetická náročnost. Oproti klasickému displeji není spotřebovávána energie při zapnutém stavu displeje, ale pouze při změně obrazu. Při statickém obrazu tedy žádnou energii displej neodebírání, a tohoto jevu využíváme často v IOT zařízeních, kde je kladen vysoký důraz na spotřebu energie.

Elektronický papír principem připomíná klasický papír - tedy zobrazení čteného textu (resp. jeho kontrast) je dáno odrazem dopadajícího světla.

Technologicky je displej umístěn na TFT matrici (Thin Film Transistor) a je složen z mikrokapslí (průměr okolo 100 mikronů). Každá mikrokapsle je naplněna bezbarvým olejovitým médiem se stovkami částic černého a bílého pigmentu. Každý typ pigmentu je nabit jiným nábojem - bílý je nabit pozitivně a černý negativně. Díky změně elektrického pole ve spodní části mikrokapsle přitahuje pigment opačného náboje a tudíž dochází ke změně barvy.[8]



Obrázek 15: Princip zobrazení na E-ink

Legenda ke schématu:

1. Vrchní vrstva
2. Transparentní vrstva elektrod
3. Transparentní mikrokapsle
4. Kladně nabitě černé kousky pigmentu
5. Záporně nabitě bílé kousky pigmentu
6. Transparentní olejová vrstva



7. Spodní vrstva elektrod, odpovídající pixelům
8. Spodní podpůrná vrstva
9. Dopadající světlo
10. Světlo odražené od bílé části
11. Světlo odražené od černé části



Obrázek 16: E-ink displej

Pro naše účely zobrazení dat byl použit monochromatický displej firmy Waveshare. Tento modul s displejem s úhlopříčkou 1,54 palců a rozlišením 200x200px byl vybrán nejen pro svou dostatečnou zobrazovací kapacitu a integrované komunikační rozhraní SPI, ale hlavně díky možnosti přepisu pouze části obrazovky. Je tedy možnost použít funkce a danou zobrazovanou část displeje rozdělit na několik dílčích částí[9] - záhlaví, navigační lištu a hlavní zobrazovací část.

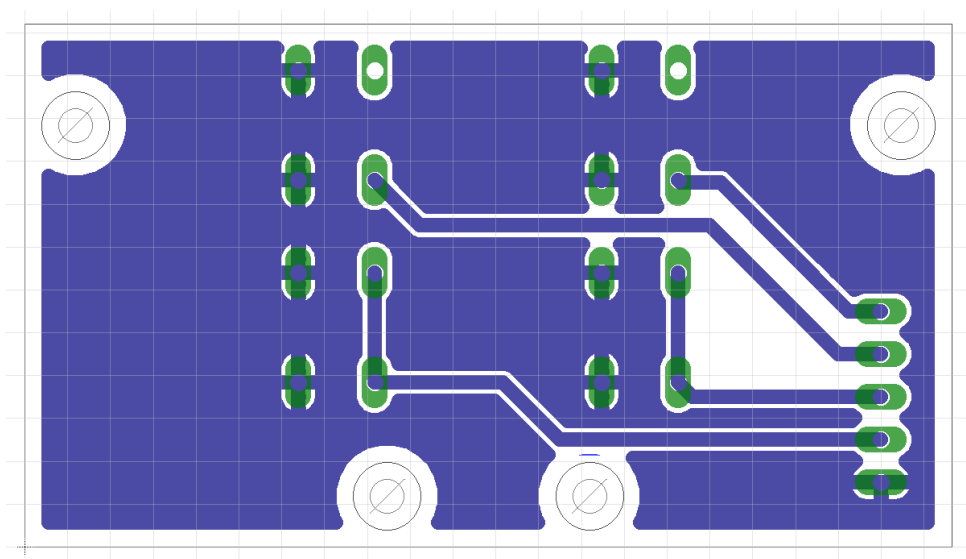
#### 5.4.2 Návrh Master node

Pro jednoduchou práci se zobrazením informací je navrženo následující rozložení: krabička je postavena na výšku, přičemž v horní části dominuje otvor pro e-ink displej a hned pod ním jsou vystouplá tlačítka sloužící k ovládání a pohybu v kontextovém menu. V tlačítkách jsou umístěny prohlavně s textem a zobrazovacími šipkami pro pohyb v menu.

Přípevnění displeje i DPS s tlačítky je pomocí šroubků (z vnitřní strany vrchního krytu).

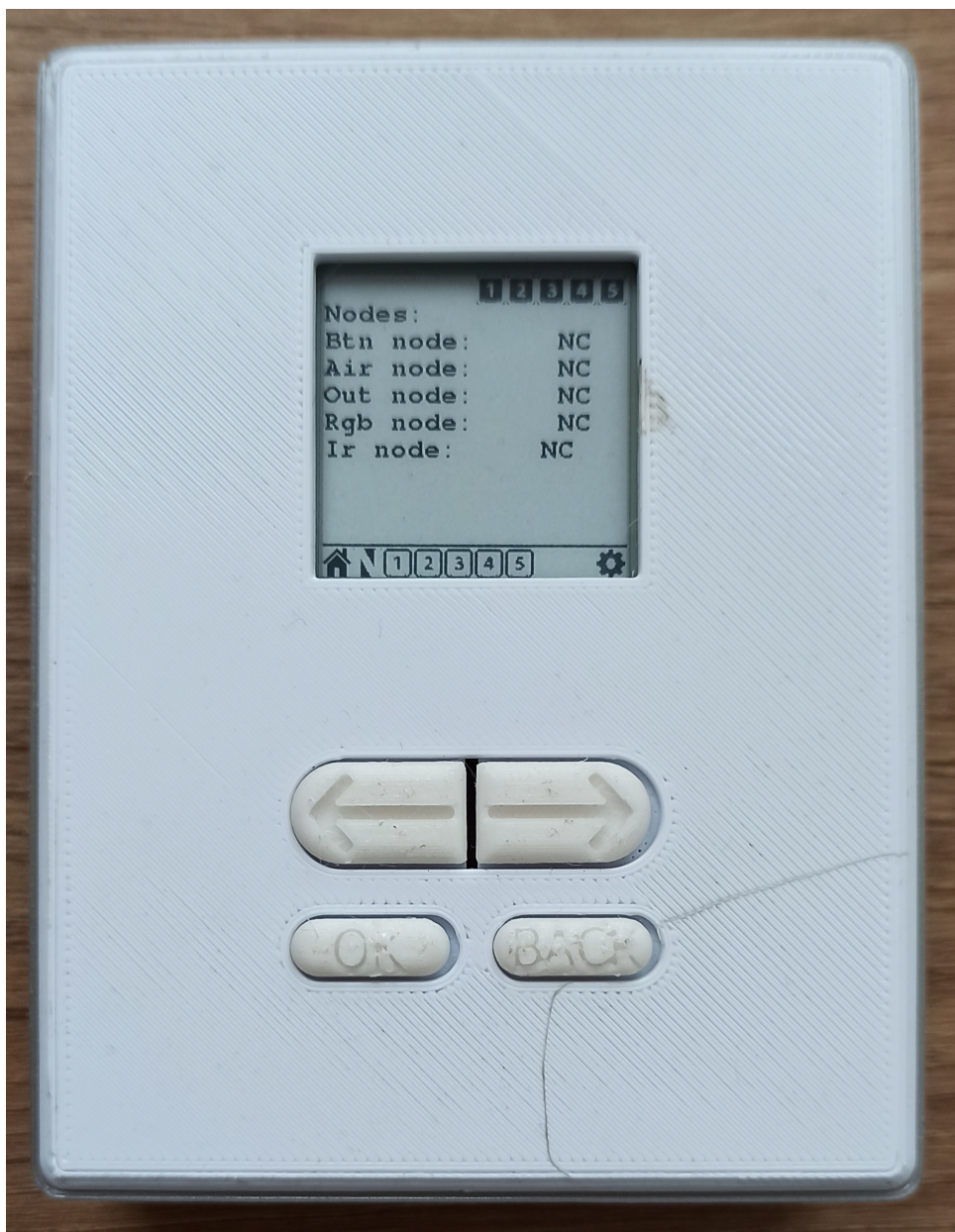


Obrázek 17: Návrh Master node



Obrázek 18: Návrh DPS pro tlačítka

### 5.4.3 Realizace Master node



Obrázek 19: Realizace Master node

### 5.5 Air node

Toto zařízení představuje prvek pro měření teploty v místosti, vzdušné vlhkosti a obsahu CO<sub>2</sub>. Tyto měřené veličiny jsou důležité především pro zajištění kvalitního života nejlépe, minimálně však pro kontrolu prostředí.



Obrázek 20: Senzorový modul SCD30

Celé zařízení tvoří ESP32 modul, napájení (Li-ion baterie a modul pro řízení energie a nabíjení) a SCD30 modul pro měření daných veličin. Spojení mezi jednotlivými komponenty je provedeno pomocí pevného drátového spojení a tudíž nebylo potřeba vytvářet DPS.

#### 5.5.1 SCD30

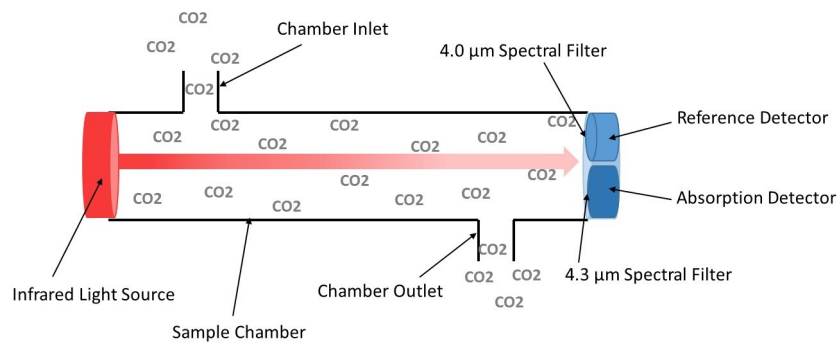
Senzorový modul SCD30 společnosti Sensirion měří teplotu, vlhkost a koncentraci oxidu uhličitého v atmosféře. Pro měření teploty a vlhkosti se na desce nachází čip SHT31 a koncentraci oxidu uhličitého zajišťuje kalibrovaný NDIR senzor se dvěma snímači. Celý modul může s řídicím mikrokontrolérem komunikovat pomocí sběrnice UART nebo I2C nebo pomocí PWM výstupu.[10]

NDIR technologie (Non Disperzive Infra-Red) je založena na fyzikálním principu absorpce oxidu uhličitého infračerveného záření určitých vlnových délek. Maximální absorpci infračerveného záření a zároveň minimální absorpci ostatních plynů má oxid uhličitý při vlnové délce  $4,3\ \mu\text{m}$ . [11]

Zdroj infračerveného záření je umístěn na jednom konci trubice, kdežto na druhém se nachází dva snímače s optickými filtry. První snímač ( $\text{CO}_2$ ) díky filtru zaznamenává záření v pásmu  $4,3\ \mu\text{m}$  a měří jeho intenzitu  $I_d$ , kdežto druhý (referenční) zaznamenává záření minimálně absorbované v okolních plynech s vlnovou délkou  $4\ \mu\text{m}$  ( $I_o$ ).

Naměřená intenzita záření na těchto vlnových délkách je pak spojena s koncentrací  $\text{CO}_2$  Beer-Lambertovým zákonem, který je vyjádřen následujícím vzorcem:

$$\frac{I_d}{I_o} = e^{-(K * C * L)}$$

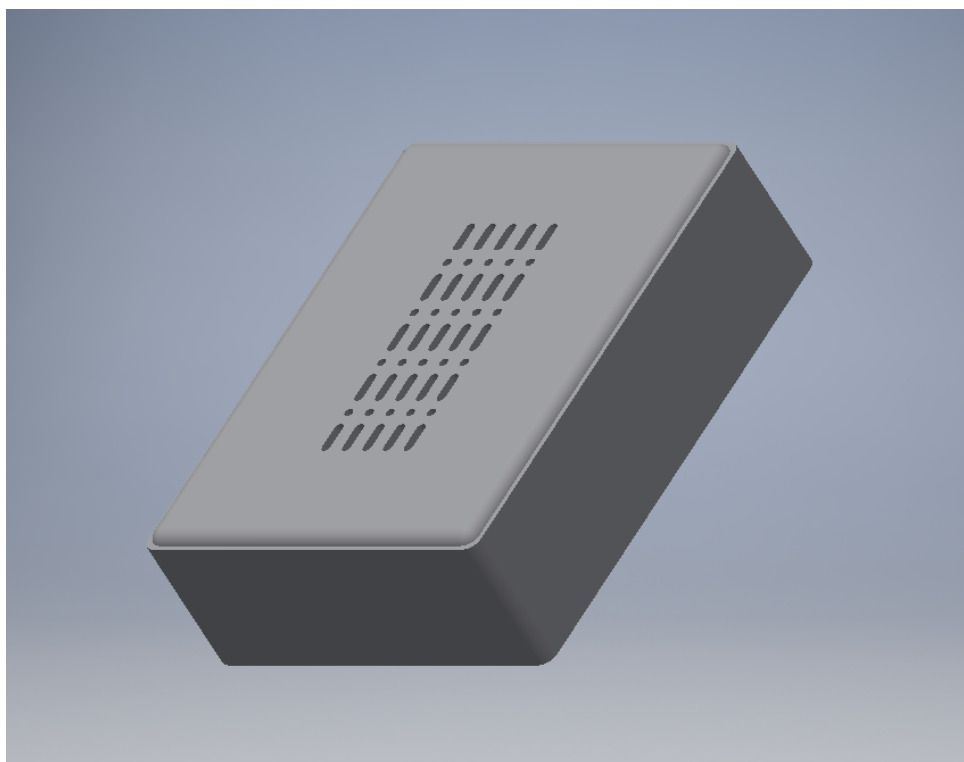


Obrázek 21: Princip NDIR

Kde  $I_d$  je intenzita záření v pásmu 4,3μm,  $I_0$  je intenzita záření v referenčním kanálu,  $K$  je koeficient absorpce pro CO<sub>2</sub>,  $C$  je koncentrace plynu a  $L$  je vzdálenost mezi zdrojem záření a senzory.

Referenční senzor kompenzuje vliv kolísání intenzity zdroje záření. Při změně intenzity se změny shodně  $I_d$  i  $I_0$ , takže poměr  $I_d/I_0$  zůstane nezměněn.

### 5.5.2 Návrh Air node



Obrázek 22: Návrh Air node



### 5.5.3 Realizace Air node

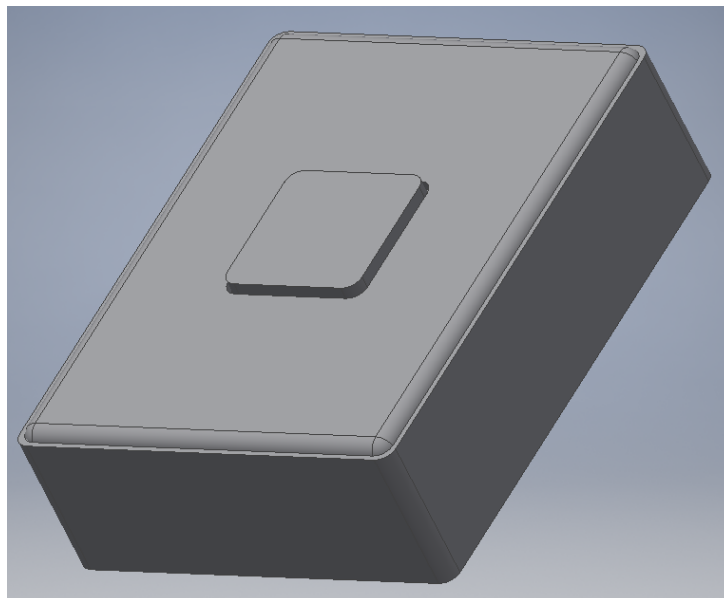


Obrázek 23: Realizace Air node

### 5.6 Btn node

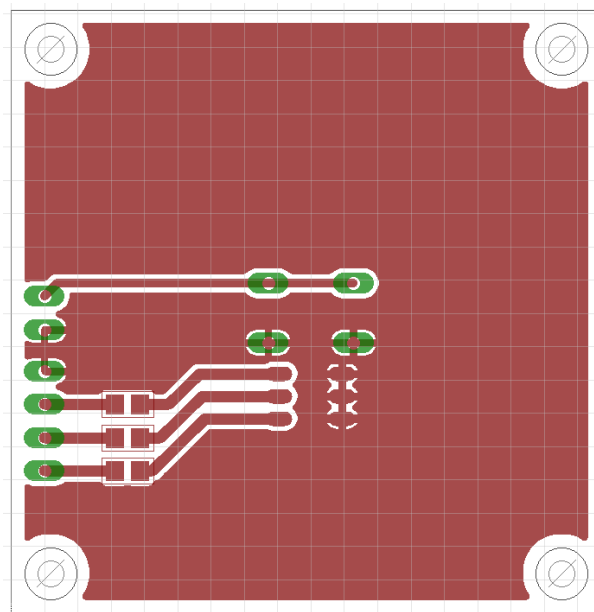
Účelem této nody je odeslat signál při stisknutí tlačítka. Pro tento účel byl vytvořen jednoduchý design - uprostřed hlavní desky je umístěn prostor pro tlačítko. Tlačítko je kryto krytkou z průsvitného plastu - pro detekování správné funkce je zde umístěna i rgb dioda.

### 5.6.1 Návrh Btn Node



Obrázek 24: Návrh Btn node

Ovládací prvek - tlačítko - je na jednostrané desce DPS. Jednoduché připojení tlačítka k připojovacím pinům a taktéž připojení indikační RGB LED diody přes rezistory. Tlačítko nemá na DPS PULLUP rezistor, protože tento rezistor je nastaven v rámci programového vybavení mikrokontroléru ESP32.



Obrázek 25: Návrh DPS pro Btn node

### 5.6.2 Realizace Btn Node



Obrázek 26: Realizace Btn node

## 5.7 Out node

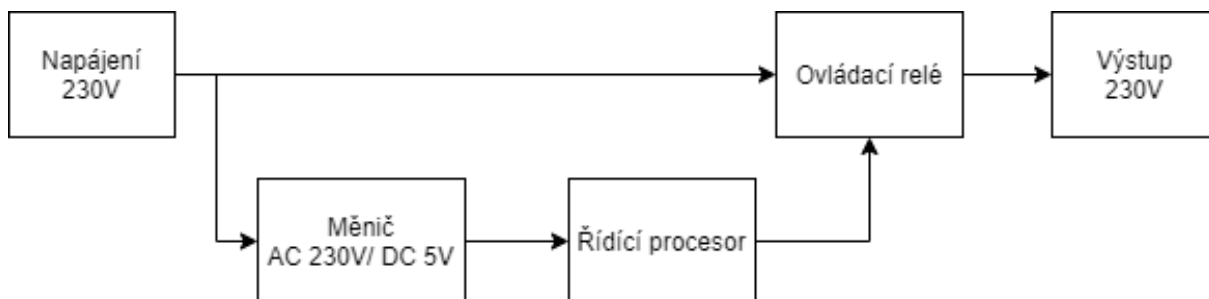
V tomto případě bylo potřeba jedinné: vytvořit dálkově ovládanou zásuvku. S tímto zařízením je možnost připojit jakékoliv síťové spotřebiče a dálkově je zapínat či vypínat. Pro ovládání slouží spárovaná zařízení - v našem případě Btn node, popřípadě Master node.

### 5.7.1 Návrh Out Node

V rámci návrhu Out node je využito stejné vrchní víčko jako u Btn node. Není to pouze vizuální potřeba, protože je zde zachována i funkce tlačítka a indikační RGB LED diody. Tlačítko v případě potřeby nemožnosti využít Btn node slouží jako vypínací/zapínací.

Ve střední části jsou provedeny úpravy pro vložení vypínače a průchodek s kabely.

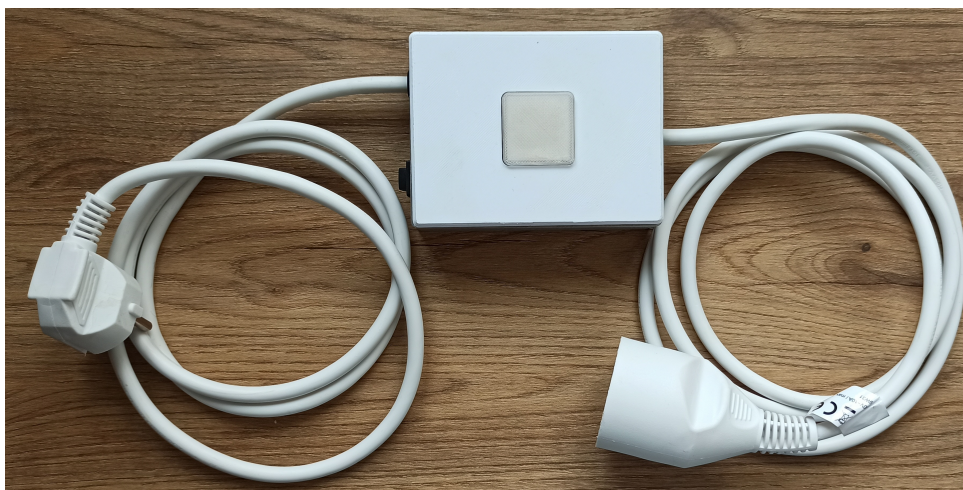




Obrázek 27: Blokové schéma Out node

### 5.7.2 Realizace Out Node

Jako ostatní zařízení, i zde je jednotný design. Veškeré potřebné věci (napájení, řídicí deska, reléový modul) jsou umístěny uvnitř. Hlavní vypínač na boku krabičky spíná proud do obvodu - tedy i napájení pro řídicí procesor. Shora je ovládací tlačítko, stejně jako na Btn node. Tlačítko slouží při odpárování Btn node jako samostatný spínač, stejně jako v případě potřeby ručního zapnutí/vypnutí.



Obrázek 28: Realizace Out node

## 5.8 Ir node

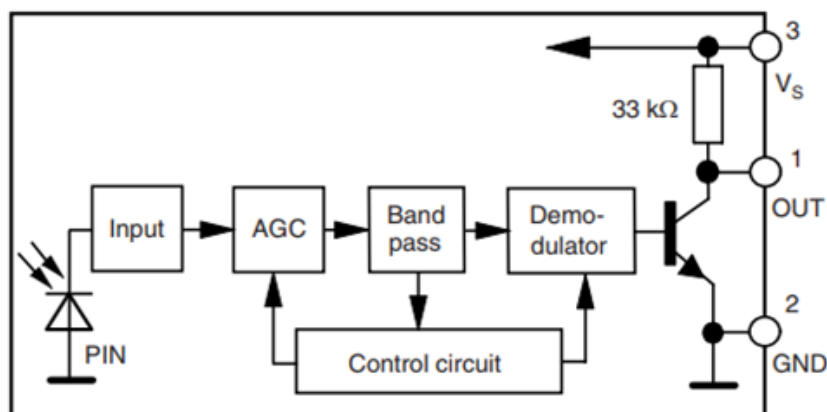
Jelikož ovládání RGB LED pásku většinou v dnešní době probíhá pomocí mobilního telefonu, je stále mnoho lidí, kteří používají k ovládání obyčejný ovladač s IR přenosem dat. Jelikož komerční zařízení mají klasicky 1 ovladač = ovládání 1 LED pásku, zde se idea samozřejmě trochu liší.

Ir node zachytává a zpracovává příchozí data(modulovaný infračervený signál), je možno nastavit v programu mnoho různých typů ovladačů. Ikdýž je Ir node koncipována k ovládání RGB LED pásku, je možno po přeprogramování použít ovladače například k zapnutí zásuvky či najít další využití.

Jelikož je zachytáván infračervený signál s jistou modulací, je potřeba tento signál demodulovat a zpracovat. K tomu slouží Ir demodulátor TSOP3848 pracující na frekvenci 38kHz.

### 5.8.1 IR demodulátor a ovladač

Tento demodulátor je vybrán především proto, že velké množství ovladačů využívá specifickou frekvenci 38kHz. Napájecí napětí demodulátoru je 2,7-5,5V, takže napájení pomocí napětí desky (3,3V) nepředstavuje žádný problém. Demodulátor má v pouzdře pullup rezistor, takže signál na výstupním pinu je log.1. Při zachycení 38kHz pulzů je signál demodulován a na výstup je přivedena log.0.



Obrázek 29: Vnitřní zapojení demodulátoru

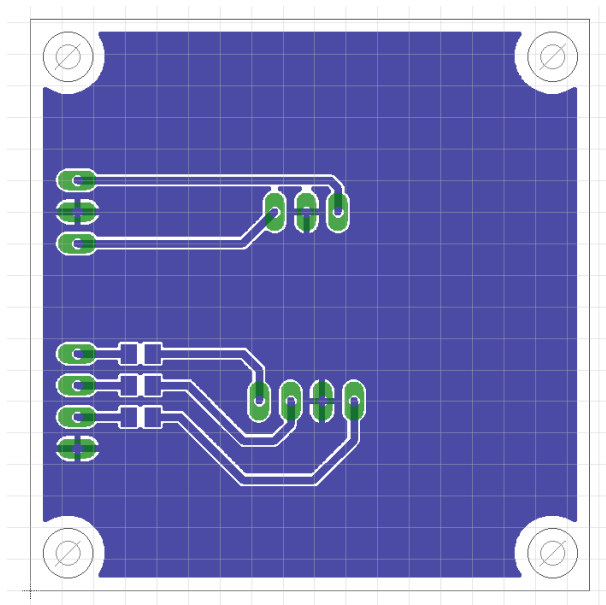
Ovladač použitý v této práci využívá této specifické frekvence pro přenos dat. Je zde možnost různých nastavení pro odeslání dat díky tomuto ovladači určenému přímo pro ovládání RGB LED pásku - přímo určené tlačítko pro stmívání, výběr jednotlivých barev nebo různých módů blikání a změny barev.



Obrázek 30: Dálkový ovladač RGB LED pásku

### 5.8.2 Návrh Ir node

V rámci návrhu Ir node je potřeba vytvořit pouze jednu desku plošných spojů. Krabíčka je s vrchním víčkem stejná jako u Btn node, ovšem místo tlačítka je zde vytvořen prostor pro IR demodulátor. Pro možnost indikace správné funkce a příjmu dat je zde také umístěna indikační RGB LED dioda.



Obrázek 31: Návrh DPS Ir node

### 5.8.3 Realizace Ir node



Obrázek 32: Realizace Ir node

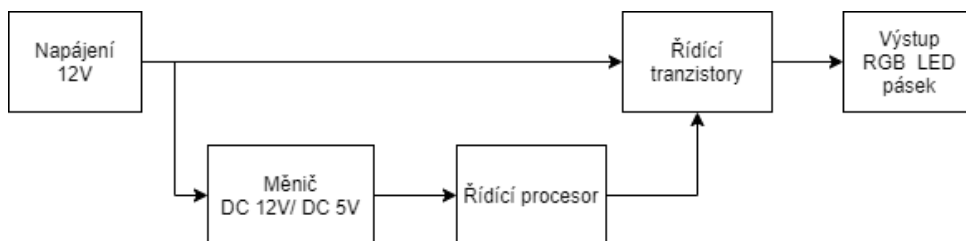
## 5.9 RGB node

Ovládání RGB LED pásku je zajištěno tímto zařízením. Jelikož většina RGB LED pásků v dnešní době používá katodové řízení, bylo potřeba vybrat i vhodné komponenty pro řízení. Ovládání samotného pásku je pomocí dvojitého tranzistoru TIP120 a pro každou barvu je určen jeden tranzistor.

### 5.9.1 Návrh RGB node

Krabička použita k uschování komponent je stejná jako u Btn node, i s použitím horního tlačítka. Tlačítko slouží v nepřítomnosti nebo nemožnosti připojit Ir node k manuálnímu vypnutí výstupů. Integrovaná RGB LED dioda u tlačítka signalizuje aktuální barvu pásku.

Napájení RGB LED pásku je pomocí stejnosměrného napětí 12V, přiváděném pomocí externího zdroje. Vnitřní modulový měnič DC/DC je nastaven na snížení napětí pro řízení mikroprocesoru ESP32.

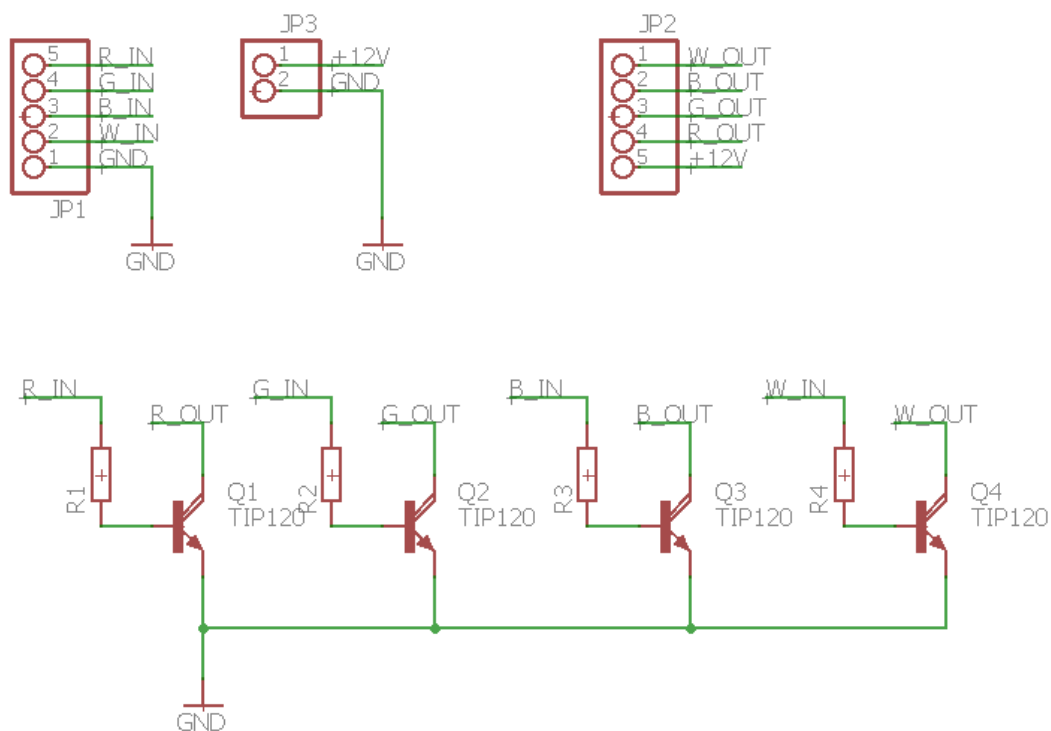


Obrázek 33: Blokové schéma vnitřního zapojení RGB node

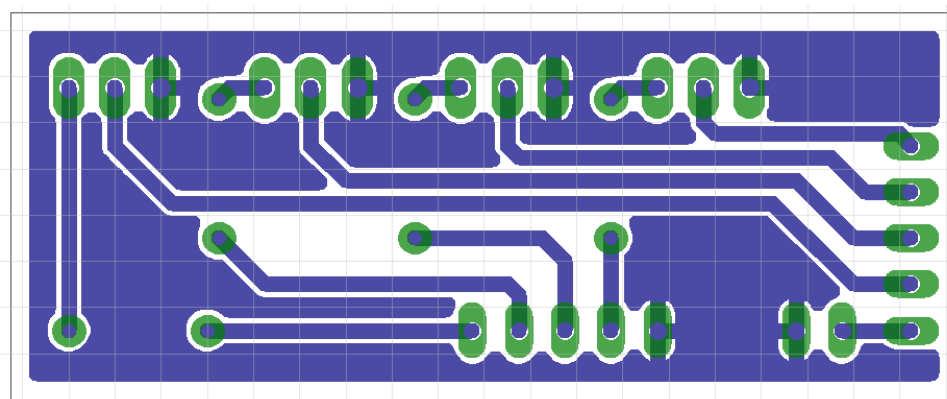
V rámci možností dalšího použití, byla vytvořena možnost připojit i RGBW pásek. Na desce plošných spojů tedy nenajdeme pouze 3 tranzistory pro jednotlivé barvy RGB, ale i čtvrtý,

který se stará o bílý kanál. Použití RGBW pásku má mnoho výhod - mezi tu největší patří jistě možnost regulovat teplotu bílého světla pomocí dodávání teplých barev z RGB části.

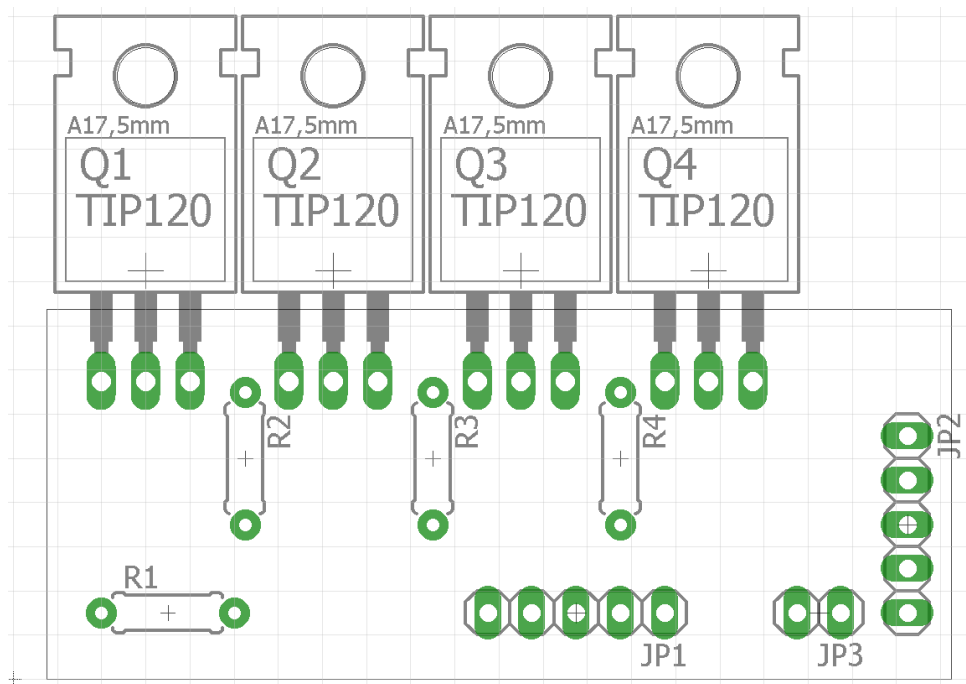
Typ řízení RGB LED pásku je řízením záporného napětí. TIP120 tranzistory jsou v pouzdře TO220 pro snazší namontování na chladič. V následujících obrázcích se nachází schématické zapojení, návrh DPS a rozmístění součástek určených pro řízení RGB LED pásku.



Obrázek 34: Návrh ovládací části pro RGB LED pásek - schéma

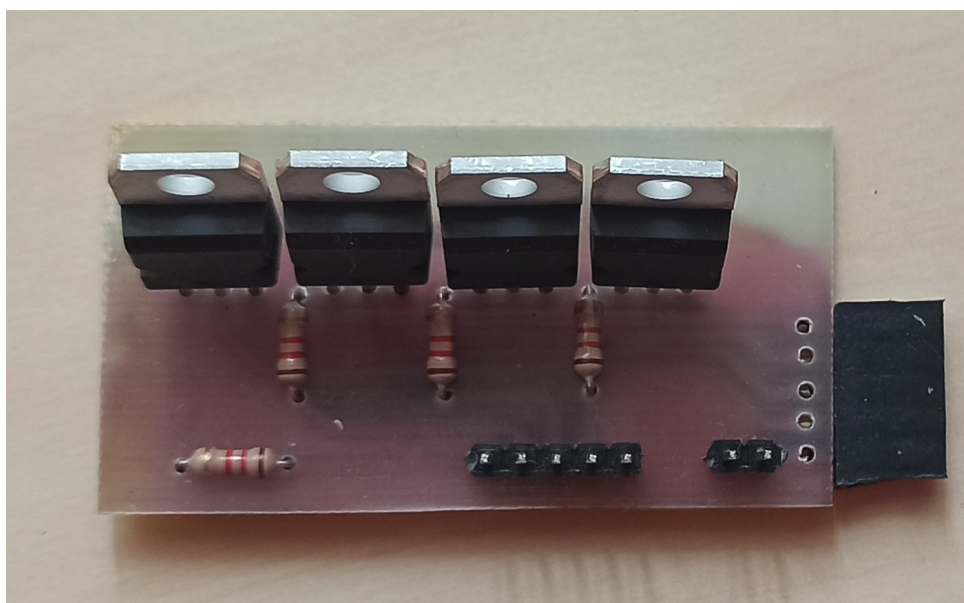


Obrázek 35: Návrh ovládací části pro RGB LED pásek - DPS



Obrázek 36: Návrh ovládací části pro RGB LED pásek - rozmístění součástek

### 5.9.2 Realizace RGB node



Obrázek 37: Realizace ovládací části pro RGB LED pásek



Obrázek 38: Realizace RGB node

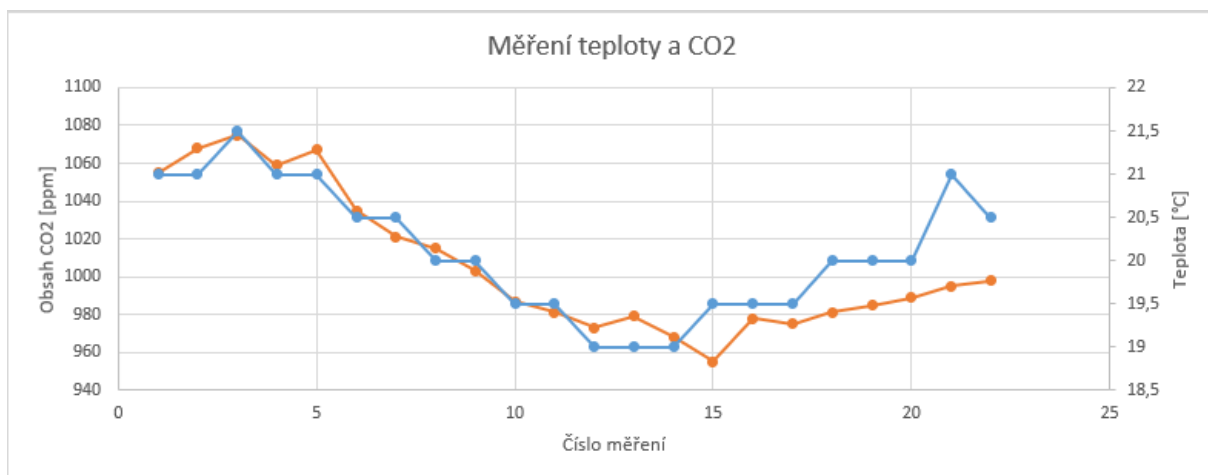
V rámci realizace RGB node byla pouze jedna chybná úvaha - a to připojení RGB LED pásku. Jelikož je řešeno připojení pomocí pin lišty - může dojít k prohození stran napájení. Proto byla vytvořena i grafická značka připojení napájení jak na krabici, tak na přívodním kabelu RGB LED pásku.



## 6 Test navrhnutého řešení

Pro finální test spojení a komunikace mezi jednotlivými zařízeními již bylo vytvořeno vše potřebné. Byly vytvořeny krabičky, do nich namontovány DPS a moduly měničů, displej, DPS s tlačítka a další komponenty. Po nahrání finální verze software byl vždy sledován výstup na jednotlivých zařízeních pomocí sériové linky.

- Air node - výsledkem testování byl výstup v podobě změřené teploty vzduchu, vlhkosti a obsahu  $\text{CO}_2$ , periodicky se opakující.



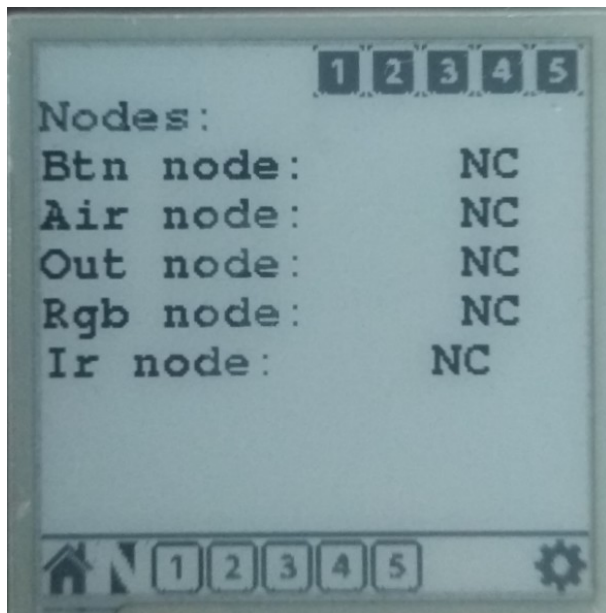
Obrázek 39: Zobrazení naměřených dat Air node

Po změření teploty došlo k vyhledávání zařízení v síti a odeslání upravených dat převedených na validní typ proměnné.

- Btn node - po stisknutí tlačítka zařízení nereagovalo nastavených 5s na další stisknutí. Ihned po stisknutí začalo prohledávat dostupné zařízení v síti a odeslalo data.
- Out node - příjem dat z Air node ani Ir node byl zachycen ale nezpracován. Při příjmu dat z Btn node byl změně stav výstupu, který sepl relé. Při příjmu dat z Master node došlo k odeslání aktuálního stavu výstupu zpět.
- Ir node - po vyslání signálu ovladačem došlo k zachycení a rozpoznání signálu, porovnání se signálem v databázi, prohledáním sítě a odesláním na připojená zařízení.
- RGB node - obdobně jakou Out node, i zde byly zachyceny data z dalších zařízení, nicméně správně zpracována pouze příchozí data z Ir node a Master node. Po přijetí signálu z Ir node došlo ke správnému nastavení barvy a módu blikání, zatímco u Master node došlo k odeslání aktuálního stavu.



- Master node - po zapojení napájení a vytvoření úvodní obrazovky bylo bez problému kontextovými tlačítky změněno menu na další obrazovku. Zde v první chvíli nebylo připojeno žádné zařízení, nicméně došlo k prohledání sítě a připojení zařízení.



Obrázek 40: Zobrazení dat na displeji

Zároveň s nalezením zařízení se změnila i ikona v hlavičce - signalizující připojené zařízení. Na obrazovkách pro jednotlivé senzory byly po připojení vždy aktuální data. Při odpojení Btn node ze systému se při aktualizaci zařízení v síti automaticky změnil stav připojení na odpojeno. Při odeslání požadavku na aktuální stav RGB node nebo Out node vždy došlo k příjmu dat s aktuálními hodnotami.

## 7 Závěr

Technologie výroby krabiček pro jednotlivá zařízení byl tisk na 3D tiskárně. V rámci této technologie byl samozřejmě kladen důraz na prototypovou výrobu vlastních navržených vizuálních stylů jednotlivých krabiček zároveň s ohledem na cenu výroby.

Přestože tisk probíhal na kvalitnějších zařízeních (3D tiskárny Průša MK3), ne vždy bylo provedeno naprosto dokonalé nastavení. Toto lze pozorovat na některých prvcích (především na vrchních víčkách a zadních krytech) kdy rozlití první vrstvy filamentu nebylo vždy naprosto dokonalé a je tedy vidět drobné čáry na těchto prvcích.

Krom těchto drobných nedostatků proběhla výroba krabiček bez problémů.

Jelikož je vnitřní vybavení jednotlivých zařízení řešeno především modulárně, nebylo ani v návrhu řešeno využití jedné desky plošných spojů pro veškeré zařízení. V rámci realizace již tedy bylo počítáno s modulem Devkit obsahujícím řídicí procesor ESP32. V případě bateriově napájených zařízení s Li-ion baterií a modulem měniče s nabíjecím obvodem, v dalších případech s modulem měničů z příslušných napětí.

Dodatečně vyráběné desky plošných spojů pro tlačítka, demodulátor a řídicí část pro ovládání RGB LED pásky byly vytvářeny v domácích podmínkách metodou fotocesty. Vytisknutí neproběhlo na fólii, nýbrž na pauzovací papír. Díky tomu (a také díky staršímu chloridu železitému) po následném vyleptání nebyly cesty ve stejné kvalitě jako po tisku v laboratorních podmínkách či dokonce ve firemních. Nicméně na výrobu prototypových desek byla tato technologie dostačující.

V rámci kompletace jednotlivých zařízení byl nejmenší problém s kompletací Master node, Air node a Btn node. U Ir node nastal problém s místem mezi DPS s demodulátorem a krytkou pro tlačítko, proto bylo potřeba vytvořit mezikus, který tento problém odstranil.

U RGB node a Out node nastal lehce problém s místem. Při skládání jednotlivých modulů dovnitř krabiček překážely spojovací dráty. U Out node byl problém se skládáním lehce obtížnější, neboť krom modulů zdroje, řídicího mikroprocesoru a relátka bylo potřeba ještě vhodným způsobem vést vstupní a výstupní dráty. Nicméně u obou zařízení se nakonec povedlo srovnat komponenty správně a bez větších problémů mohlo dojít k zavření krabiček.

Ještě jeden krok potřebný k úpravě komponent krabiček byl potřeba u bateriově napájených zařízení - a to vyříznutí otvoru pro přívod napájení k dobíjecímu modulu pro li-ion baterie. Tento otvor byl vyříznut v zadním krytu kvůli lepší manipulaci s nabíjecím kabelem (micro USB) - oproti původní myšlence (kdy pro nabíjení baterie by bylo potřeba odšroubovat zadní kryt).

Po praktické stránce výroby tedy nebyl žádný velký problém (kromě drobných stylistických vad 3D tisku, úpravy zadního krytu a správného seskládání komponent u RGB node a Out node).

V rámci spojení a správné funkčnosti po softwarové stránce to bylo trochu jinak. Instalace aktuálního Arduino IDE proběhla hladce, stejně tak jako přidání vývojové platformy ESP32. Nejprve byly programovány funkce pro správný chod jednotlivých senzorů (SCD30, Ir demodulátor, snímání přerušení tlačítkem a e-ink displej). Jediný zádrhel se objevil při nastavení

SCD30 - konkrétně doby měření. Jelikož senzor nemá být nastavený na neustále měření v intervalu 2s ale delší periody, není nastavení periody měření pomocí komunikace, ale pomocí zapnutí napájení.

Po úspěšné sprovoznění senzorů nastala část se sprovozněním komunikace. Jelikož oficiální nastavení pomocí knihoven ESP32 je half-duplex - tedy jedním směrem, bylo potřeba upravit části knihoven pro správnou funkci. Nakonec proběhlo nastavení komunikace hladce.

Zařízení dokázala nalézt ostatní v síti, spárovat a uložit do paměti. Pro jednodušší práci byla vytvořena knihovna HAS.h, které umožňuje z jednoho místa spravovat MAC adresy jednotlivých zařízení a pomáhá s úpravou odesílaných dat na správné proměnné.

Celkem tedy jednotlivé zařízení komunikují bez problému. Jediný háček je v délce odezvy, která ze vstupních zařízení (Btn node a Ir node) díky skenování sítě trvá 2s.

Variabilita a možnost úpravy systému rozšířením je obrovská. S možností ovládat díky integrovanému bluetooth modulu zařízení přes chytrý telefon nebo tablet, či prostě přes internet, ukládat data na cloud nebo jen přidat další senzory a aktuátory dává další možnosti rozšíření.

## Literatura

1. *Co je to IOT* [online] [cit. 2020-01-10]. Dostupné z: <https://www.iot-portal.cz/co-je-iot/>.
2. *Systémová technika budov* [online] [cit. 2020-01-12]. Dostupné z: <http://fei1.vsb.cz/kat420/vyuka/TZB/systemova%20technika%20budov.pdf>.
3. *Instalace desky ESP32* [online] [cit. 2020-02-03]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/jednoducha-instalace-esp32-do-arduino-ide.html>.
4. *Dokumentace api ESP-NOW* [online] [cit. 2019-12-25]. Dostupné z: [https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp\\_now.html](https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp_now.html).
5. *Konverze mezi modely barev* [online] [cit. 2020-01-15]. Dostupné z: <https://www.vocal.com/video/rgb-and-hsvhsihs-l-color-space-conversion/>.
6. *ESP32 Datasheet* [online] [cit. 2019-11-10]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
7. *Datasheet MT3608* [online] [cit. 2020-03-15]. Dostupné z: <https://drive.google.com/file/d/1-fcE-E0KfDZti8R26tSqry23xS1-4Vmv/view>.
8. *E-ink technologie* [online] [cit. 2020-03-20]. Dostupné z: <http://noel.feld.cvut.cz/vyu/a2b31hpm/index.php/U%C5%BEivatel:Subcizde>.
9. *E-ink displej* [online] [cit. 2020-03-15]. Dostupné z: <https://www.waveshare.com/product/1.54inch-e-Paper-Module.htm>.
10. *Datasheet SCD30* [online] [cit. 2020-03-17]. Dostupné z: [https://cz.mouser.com/datasheet/2/682/Sensirion\\_CO2\\_Sensors\\_SCD30\\_Datasheet-1510853.pdf](https://cz.mouser.com/datasheet/2/682/Sensirion_CO2_Sensors_SCD30_Datasheet-1510853.pdf).
11. *Technologie NDIR* [online] [cit. 2020-03-08]. Dostupné z: <https://www.soselectronic.cz/articles/sensirion/scd30-je-vice-nez-ndir-co2-senzor-2152>.

## A Seznam příloh

Součástí práce jsou přílohy v elektronické podobě.

- Zdrojaky
  - Air-node.ino.elf
  - Btn-node.ino.elf
  - Ir-node.ino.elf
  - Master-node-tasks.ino.elf
  - Out-node.ino.elf
  - HAS (složka se soubory knihovny)
- 3D tisk
  - btn-krytka.stl
  - master-tlacitka.stl
  - stred-univ.stl
  - vicko-air.stl
  - vicko-btn.stl
  - vicko-master.stl
  - zadni-kryt.stl
- Schemata
  - btn-btn.brd
  - btn-btn.sch
  - Ir.brd
  - Ir.sch
  - master-btn.brd
  - master-btn.sch
  - RGB.brd
  - RGB.sch